

AD-A237 922



NWC TP 7075

2

Analog Design Verification With PSpice and MICRO-CAP III

by
Melody Witham
Attack Weapons Department

AUGUST 1990

NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555-6001



DTIC
ELECTE
JUL 10 1991
S B D

Approved for public release; distribution is unlimited.

91 7 8 020

91-04329



Naval Weapons Center

FOREWORD (U)

Computer simulation to verify hardware design is becoming more popular. This report documents the design of two circuits, their results from the lab, and their simulation on two computer programs: PSpice and MICRO-CAP III. The use of PSpice is also explained. These investigations were performed at the Naval Weapons Center and were carried out under Air Task no. A540-5402/008-C/OW/8070000.

This report was reviewed for technical accuracy by Richard Smith Hughes.

Approved by
P. B. HOMER, *Head*
Attack Weapons Department
22 June 1990

Under authority of
D. W. COOK
Capt., U.S. Navy
Commander

Released for publication by
W. B. PORTER
Technical Director

NWC Technical Publication 7075

Published by.....Attack Weapons Department
Collation.....Cover, 29 leaves
First printing.....125 copies

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| | | | |
|---|---|--|---|
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE August 1990 | 3. REPORT TYPE AND DATES COVERED Summary; January-March 1990 |
| 4. TITLE AND SUBTITLE Analog Design Verification With PSpice and MICRO-CAP III | | | 5. FUNDING NUMBERS |
| 6. AUTHOR(S) Melody Witham | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Weapons Center China Lake, CA 93555-6001 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER NWC TP 7075 |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |
| 11. SUPPLEMENTARY NOTES | | | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. | | | 12b. DISTRIBUTION CODE |
| 13. ABSTRACT (Maximum 200 words) The operation of PSpice is explained. The capabilities of PSpice and MICRO-CAP III are discussed. Two example circuits were designed, built, and tested in the lab and on the computer using PSpice and MICRO-CAP III. Calculations done by hand are compared to those obtained in the lab and on the computer. | | | |
| 14. SUBJECT TERMS Analog circuit analysis Computer aided design (analog) Differential amplifiers MICRO-CAP III PSpice Video amplifiers | | | 15. NUMBER OF PAGES 55 |
| | | | 16. PRICE CODE |
| 17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED | 20. LIMITATION OF ABSTRACT SAR |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

CONTENTS

| | |
|--|----|
| Nomenclature | 2 |
| Introduction | 5 |
| Using PSpice..... | 6 |
| Creating a Circuit Input File | 6 |
| Running PSpice | 11 |
| Using the Control Shell..... | 12 |
| Using Probe | 16 |
| The Stimulus Editor (StmEd) | 19 |
| Capabilities of PSpice and MICRO-CAP III..... | 19 |
| Comparison of the Two Programs..... | 19 |
| Example Circuits..... | 24 |
| Differential Amplifier | 24 |
| Video Amplifier | 36 |
| Comparison of Results..... | 41 |
| Differential Amplifier (Transistor) | 42 |
| Differential Amplifier (Chip)..... | 42 |
| Video Amplifier (Chip) | 43 |
| Video Amplifier (Transistor)..... | 44 |
| Conclusion | 44 |
| References..... | 45 |
| Bibliography | 46 |
| Appendixes: | |
| A. Test Setup | 47 |
| B. PSpice Input File..... | 49 |
| C. MICRO-CAP III Circuit Diagram..... | 53 |

NOMENCLATURE

| | |
|--------|--|
| .AC | AC analysis command (PSpice) |
| ACCT | statistics summary (PSpice, .OPT command) |
| ASCII | American Standard Code Information Interchange |
| C | capacitor |
| CLOAD | capacitor for the differential amplifier |
| CRES | resistor model for the differential amplifier |
| .DC | DC analysis command (PSpice) |
| DEC | decade |
| DEV | individual device tolerance (PSpice, .MODEL statement) |
| DOS | disk operating system |
| .FOUR | command for Fourier analysis (PSpice) |
| I | current |
| k | kilo |
| .LIB | statement to point to device library (PSpice) |
| LOT | lot tolerance (PSpice, .MODEL statement) |
| .MC | command for Monte Carlo analysis (PSpice) |
| .MODEL | command to specify device models (PSpice) |
| .NOISE | command for noise calculation (PSpice) |
| NOMOD | no model parameters (PSpice, .OPT command) |
| NOPAGE | no pagination (PSpice, .OPT command) |
| .OP | small-signal parameters output command (PSpice) |
| .OPT | options command (PSpice) |

NWC TP 7075

.PARAM command that assigns parameters' values (PSpice)
 .PLOT command for output in graphical format (PSpice)
 .PRINT command for output in tabular format (PSpice)
 PROBE PSpice's post-processor
 PS command to enter PSpice's Control Shell
 PS -M command to enter PSpice's Control Shell with altered graphics

 QNL transistor model for differential amplifier

 R resistor
 RELTOL relative accuracy (PSpice, .OPT command)
 R_T resistance at a specific temperature

 TC1 linear temperature coefficient (PSpice, .MODEL statement)
 TC2 quadratic temperature coefficient (PSpice, .MODEL statement)
 .TEMP temperature command (PSpice)
 .TF transfer function analysis command (PSpice)
 .TRAN command for transient analysis (PSpice)

 V voltage
 VCC positive supply voltage
 VEE negative supply voltage
 V_{IN} input voltage

 .WIDTH number of columns for PSpice output

 YMAX greatest difference between waveform and nominal calculated
 (PSpice, .MC command)

| | |
|--------------------|-------------------------------------|
| Accession For | |
| NTIS GRA&I | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

INTRODUCTION

Computer simulation has become very important in the design world. Few engineers design a circuit and send it right to the shop to be built. Instead, some level of simulation is used. Because fewer prototype boards need be built, simulation saves a designer many hours of debugging time in the lab and quite a bit of money.

Digital simulators are considered to be very accurate and indispensable tools. However, analog and mixed analog and digital simulators have not been perfected. With a digital chip, everything is black and white (or one and zero). Only the timing raises some questions. With an analog circuit, the model is one of the most important items. Without a good device model, the simulation is worthless.

Two bipolar models are most popular: Gummel-Poon and Ebers-Moll. Mathematically, the models are similar, except that the Ebers-Moll model ignores second-order effects but gives a good first-order model of terminal currents and charge storage effects. The Gummel-Poon model takes base width modulation, high-level injection, and base-widening into account. The Gummel-Poon model should be the model of choice if very accurate results are needed. If this is not the case, then the Ebers-Moll model might not only be sufficient but also decrease the simulation time.

PSpice* uses an "enhanced Gummel-Poon" model, while MICRO-CAP III (MC3) can use either the Ebers-Moll or the Gummel-Poon model. Most of the simulations documented here used the Gummel-Poon model to more closely relate to PSpice results. More information about the PSpice model can be found in Reference 1, starting on page 96. Information about the MICRO-CAP III models can be found in Reference 2, starting on page 3-1.

Learning to use a simulator takes some time but pays off in the end. MICRO-CAP III is very easy to learn to use and the manuals are fairly good. PSpice is a little more difficult to learn and the manual is fair. To aid the first time user, an explanation of the use of PSpice is included below.

* All company/product names are trademarks/registered trademarks of their respective holders.

USING PSPICE

PSpice is different from many simulators being used today because of the way circuit information is input into the computer. Most simulators use an interactive schematic capture input program with the circuit drawn for ease of visualization. PSpice does not have schematic capture, so the user must specify node numbers. (It is hard to visualize a PSpice circuit by just looking at the input file). Also, if an element is added to the circuit, node numbers may have to be changed to accommodate the new component. Once the operation of PSpice is learned, however, it is fairly easy to operate and make changes. The Control Shell makes PSpice more user friendly to the beginner but can become cumbersome with continued use. It is easier to use a favorite word processor and then run PSpice from the DOS prompt. Both methods, however, will be explained.

CREATING A CIRCUIT INPUT FILE

The input file can have any file name, as long as it has the extension `.cir`. The first line in the circuit file must be the title line and can contain any text that the designer wants. Figure 1 is an example of a circuit file. The last line in the circuit file is `.END`, also shown in Figure 1. The rest of the lines in the circuit file can be in any order desired. They do not have to be in the order explained in this text, but the file will be easier to read if it is logically arranged.

The second line in the example circuit file is the `.OPT` statement. This command specifies the options for the analyses. `ACCT` orders a job statistics summary at the end of the output file, including a counting of nodes for several different cases, a time summary in seconds, and iterations for each of the simulation steps. `NOMOD` suppresses the listing of model parameters and temperature updated values. `NOPAGE` stops pagination after each major section and suppresses the banners for each major section. `RELTOL=.001` gives the relative accuracy of V_s and I_s . There are also other options for this command. A description of each option can be found in Reference 1, starting on page 137.

The `.WIDTH` statement orders the number of columns (80 in Figure 1). The only other possible configuration is 132 columns. If the `.WIDTH` statement is not included in a file, then the default value is 80 columns.

The next statement in Figure 1, `.TEMP`, sets the run temperature at 35°C. If other temperatures are specified, then an analysis will be run for each temperature.

NWC TP 7075

```

DIFFAMP -- Differential Amplifier Circuit
.OPT ACCT NOMOD NOPAGE RELTOL=.001
.WIDTH OUT=80
.TEMP 35
.DC LIN VIN -0.125 0.125 0.005
.OP
.TF V(5) VIN
.AC DEC 10 100KHZ 10GHZ
.NOISE V(5) VIN 30
.TRAN 20NS 1000NS
.FOUR 5MEG V(5)
.MC 5 DC V(4,5) YMAX
.PARAM FACTOR=1.2
VIN 100 0 AC 1 SIN(0 0.1 5MEG)
VCC 101 0 DC {10*FACTOR}
VEE 102 0 {-10*FACTOR}
Q1 4 2 6 QNL
Q2 5 3 6 QNL
RS1 100 2 1K
RS2 3 0 1K
RC1 4 101 CRES 10K
RC2 5 101 CRES 10K
Q3 6 7 102 QNL
Q4 7 7 102 QNL
RBIAS 7 101 20K
CLOAD 4 5 5PF
.MODEL CRES RES (R=1 DEV=5% TC1=.02 TC2=.0045)
.MODEL QNL NPN (BF=80 RB=100 CCS=2PF TF=0.3NS TR=6NS
+   CJE=3PF CJC=2PF VA=50)
.PRINT DC V(4) V(5)
.PLOT DC IC(Q2)
.PLOT AC VCM(Q2) VCP(Q2)
.PLOT NOISE INOISE ONOISE
.PLOT TRAN V(4,5) V(10)
.END

```

FIGURE 1. PSpice Input Circuit File.

The next statements command that different analyses be performed. The first is the **.DC** statement. This command calculates the circuit's bias point over a range of values and performs a small-signal bias point calculation using the nonlinear device

equations. VIN is swept from -0.125 to 0.125 volt and the step size is 0.005 volt. In this case a linear sweep is performed; however, it is possible to request a logarithmic sweep or a sweep performed from a list of values. Reference 1, page 120, contains more information on these statements.

The **.OP** command outputs the small-signal parameters for each device and the currents and power dissipations of all the voltage sources. If no **.OP** statement is specified in a file, then the bias point is calculated, but only a list of the node voltages is generated in the output file.

The small-signal transfer function is calculated by the **.TF** statement. VIN is the input and V(5) is the output.

The **.AC** statement calculates the frequency response of the circuit over a range of frequencies. The sweep in the example is calculated in decades (**DEC**) but it can also be calculated in octaves or linearly. The next number (10) is the number of points in the sweep, or in each decade or in each octave. The next two numbers indicate that the sweep is performed between 100 kHz and 10 GHz. To get AC results in the output file, a **.PRINT**, **.PLOT**, or **.PROBE** statement must be used.

Noise calculations are performed during the AC analysis with the **.NOISE** statement. Because the noise analysis is performed in conjunction with the AC analysis, there must be an **.AC** statement. The noise is calculated and propagated to node 5 (V(5) in Figure 1). The input noise is calculated at VIN. The output includes an output noise printout for every frequency; and every thirtieth frequency, a detailed table of each device's contribution to the noise is outputted.

The **.TRAN** statement causes a transient analysis to be performed. The circuit's behavior over time is calculated from 0 to 1000 nanoseconds (ns) in print step intervals of 20 ns. The transient analysis uses an internal time step, which is adjusted according to the activity level. Then the results are printed at the time interval specified. A second order polynomial interpolation calculates these time-interval-specified values from the values calculated at the times selected by the computer. **.PRINT**, **.PLOT**, or **.PROBE** is used to get the results of the analysis. See Reference 1, page 149, for more information on the **.TRAN** statement.

A Fourier analysis is performed on the waveform V(5) during the transient analysis delineated at the **.FOUR** statement. The fundamental frequency is 5 MHz. The magnitude and phase of the fundamental and of the first eight

harmonics are calculated and automatically placed in the output file. (In **.Probe**, however, a complete spectra is displayed.)

The **.MC** statement performs a Monte Carlo analysis. In Figure 1, the DC sweep is performed 5 times using the tolerances in the **.MODEL** statement, comparing the waveform **V(4,5)** calculated in each run with the nominal **V(4,5)**. A table of each run's deviation from the nominal is created. **YMAX** is used to find the greatest difference between the waveform and the nominal. Other functions may be specified if desired. Reference 1, page 130, contains more information on **YMAX**.

The next statements describe circuit operation. The **.PARAM** statement allows parameters to be assigned and used throughout the circuit file. For the circuit file in Figure 1, the parameter **FACTOR** is given a value of 1.2 (which sets the value of the supply voltages).

An item that starts with a **V** is used to specify a voltage. In Figure 1, **VIN** is a voltage input between nodes 100 and 0 (0 is always ground). **VIN** has an AC input of 1, a transient input that is a sine wave with 0 offset voltage, an amplitude of 0.1 volt, and a frequency of 5 MHz. **VCC** is a DC voltage that is 10 times the value of **FACTOR** in the parameter statement and is placed between nodes 101 and 0. **VEE** is a DC voltage placed between nodes 102 and 0 that has a value of -10 times **FACTOR**.

The NPN bipolar transistors start with a **Q**, and there are four transistors in the circuit file of Figure 1. The first is **Q1**. **Q1**'s collector is tied to node 4, its base to node 2, and its emitter to node 6. All these transistors refer to the model **QNL**, which is specified later in the circuit file in the **.MODEL** statement.

Resistors' names start with an **"R"**. There are five resistors in this circuit: **RS1**, **RS2**, **RC1**, **RC2**, and **RBIAS**. The node names the resistors are placed between are the first two numbers in the statement. For example, **RS1** is between nodes 100 and 2. The next number is the value of the resistor. **RS1** has a value of 1 k Ω . If the next item is not a number specifying the value of the resistor, it is a model name with the value of the resistor following it. **RC1** is a 10 k Ω resistor following the **CRES** model. **CRES** is defined with the **.MODEL** statement later in the circuit file.

A capacitor is specified with a name starting with a **"C"**. In Figure 1, **CLOAD** is a capacitor between nodes 4 and 5 that has a value of 5 picofarads (pF). Models can also be used with capacitors.

NWC TP 7075

There are many other circuit items used in PSpice, and more information about them and those described above can be found in Table 1 and in Reference 1.

TABLE 1. PSpice Circuit Devices.

| PSpice notation | Device | Reference 1 page no. |
|-----------------|---|----------------------|
| B | GaAsFET | 46 |
| C | Capacitor | 51 |
| D | Diode | 52 |
| E | Voltage-controlled voltage source | 55 |
| F | Current-controlled current source | 57 |
| G | Voltage-controlled current source | 58 |
| H | Current-controlled voltage source | 60 |
| I | Independent current source and stimulus | 61 |
| J | Junction FET | 69 |
| K | Inductor coupling (transformer core) | 73 |
| L | Inductor | 77 |
| M | MOSFET | 78 |
| N | Digital input | 88 |
| O | Digital output | 92 |
| Q | Bipolar transistor | 96 |
| R | Resistor | 102 |
| S | Voltage-controlled switch | 104 |
| T | Transmission line | 106 |
| U | Digital device | 107 |
| V | Independent voltage source and stimulus | 108 |
| W | Current-controlled switch | 116 |
| X | Subcircuit call | 118 |

The **.MODEL** statement is used to define the parameters of several devices within a circuit. The first **.MODEL** statement is for **CRES**, a resistor, indicated by **RES**. The values in parentheses specify the parameters of the model. **R** is the resistance multiplier, in this case 1. **DEV** indicates the individual device tolerance. In this example, all the resistors with the **CRES** model will be within 5% of their

specified value. Each value, however, is independent of any other. If a **LOT** tolerance was specified, all the values would have to be within a certain percentage of each other. **TC1** is the linear temperature coefficient and **TC2** is the quadratic temperature coefficient. These coefficients are used to calculate the effect that temperature has on the value of the device. The equation to calculate the resistance (R_T) is

$$R_T = < \text{value} > * R * \left(1 + TC1 * (T - T_{nom}) + TC2 * (T - T_{nom})^2 \right) \quad (1)$$

<value> is the value of the resistor stated in the individual resistor statement. For more information on the equations for the temperature coefficients, see Reference 1, page 102.

The second **.MODEL** statement is for the transistors using the **QNL** model. These transistors are NPN devices, with their parameters included in parentheses. For an explanation of these parameters see Reference 1, page 96. To use a model that is in a library, put the name of the device in the same place as the model name in the device statement (in place of **QNL** in the **Q1** statement). Then use the **.LIB** statement to specify which directory contains the component. For example, the 2N2222A transistor is found in the **C:\PSPICE\BIPOLAR.LIB** library. See Reference 1, page 129, for more information on the **.LIB** statement.

The following commands provide output in both graphical and tabular form for specified voltages and currents. To obtain results in a tabular form, use the **.PRINT** statement; to obtain them in graphical form, use the **.PLOT** statement. Results can be taken from the **DC**, **AC**, **NOISE**, and **TRAN** analyses. The analysis is chosen by placing its name after the **.PRINT** or the **.PLOT** command; then the voltages and currents are added. Figure 1 contains several examples of these commands.

The last statement of the file is the **.END** statement. Explanations of other commands can be found in Reference 1. Only the most important commands were explained here.

RUNNING PSPICE

PSpice is run by the command

PSPICE filename

A status window informs the user of the run's progress. The window will disappear at the end of the run and the prompt will return. To view the results, the user types or prints the output file. The output file should have the same name as the input file, but with a **.OUT** extension.

USING THE CONTROL SHELL

The Control Shell can be used through the whole PSpice process, from creating the input file to viewing the results. The Shell is called up with the command

PS

If this command brings up a blank screen and the graphics do not seem to be working correctly, hit the right arrow key two times, then hit **Return** two times. This should return the computer to the prompt. Then type the command

PS -M

At this point a blank screen with menus at the top and bottom should appear.

The first menu heading is **Files**. Hit **F** or move the cursor to this heading and hit **Return**. The following subheadings appear:

Edit
Browse
Current
Save
Xedit

Hit **C** or move the cursor to **Current** and hit **Return**. Type in the name of the file desired or the name of the new file. To get a list of files available, hit **F4**. Move the cursor to the file desired and hit **Return**. If there are no syntax errors, the file will be loaded; if there are errors, a message will appear. Hit **F6** and a list of the errors will be displayed. Hit **PgUp** and **PgDn** to move in the window and hit **F6** to close the window.

To edit the file, hit **E** or move the cursor to **Edit** and hit **Return**. Hit **ESC** to end editing and **S** to save or **D** to discard changes.

To view the output file, hit **B** or move the cursor to **Browse** and hit **Return**. To close the **Browse** window, hit **ESC**.

To save the current file, hit **S** or move the cursor to **Save** and hit **Return**.

The last item in the Files menu is **Xedit**. This is an external editor intended for running a schematic capture editor. In most cases, it will be easier to use the regular **Edit**.

The next main menu heading is **Circuit**. It contains the following subheadings:

Devices
Models
Parameters
Errors

The **Devices** menu allows the user to change device values without going into the **Edit** menu. To select **Devices**, hit **D** or move the cursor to **Devices** and hit **Return**. A list of devices and their parameters will appear. Select the desired device by moving the cursor to that device and hitting **Return**. A list of that device's parameters will appear. Move the cursor to the parameter desired and hit **Return**. Make the necessary change and hit **Return** again. Hit **ESC** and see that the change has been made. Hit **ESC** again to get out of the **Devices** menu.

The **Models** menu allows changes to be made to model parameters. To enter the **Models** menu, hit **M** or move the cursor to the **Models** menu and hit **Return**. Move the cursor to the desired model and hit **Return**. Move the cursor to the parameter that needs to be changed and hit **Return**. Make the change to the parameter and hit **Return**. Hit **ESC** to return to the list of models and hit **ESC** again to exit the **Models** submenu.

To change parameter values, select the **Parameters** menu by hitting **P** or moving the cursor to **Parameters** and hitting **Return**. Move the cursor to the parameter that needs to be changed and hit **Return**. Type in the new value and hit **Return**. Hit **ESC** to exit.

The **Errors** submenu lists syntax errors. Hit **F6** or move the cursor to **Errors** and hit **Return**.

The next main menu is for the **Stimulus Editor**. To use the Stimulus Editor menu, hit **S** or move the cursor to **StmEd** and hit **Return**. There are only three items in this submenu:

Edit
Command File
Log to File

To enter the Stimulus Editor, hit **E** or move the cursor to **Edit** and hit **Return**. The operation of the Stimulus Editor is not explained in this report. See Reference 1, page 193, for more information.

To specify whether a command file is to be used, hit **C** or move the cursor to **Command File** and hit **Return**. Specify **Y** (yes) or **N** (no) and the filename and hit **Return**. Two small arrows in front of a command signify that that command is activated.

To specify whether a log file should be generated, hit **L** or move the cursor to **Log to File** and hit **Return**. Specify **Y** or **N** and the filename and hit **Return**.

The next main menu is the **Analysis** menu. It contains the following items:

Start
AC & Noise
DC Sweep
Transient
Run Temp
Monte Carlo
Change Options

To have PSpice run the analyses selected (shown by double arrows), hit **S** or move the cursor to **Start** and hit **Return**. This will run PSpice as if the Control Shell was not being used. After the run is complete, the user will be put into **Probe**, and then after exiting **Probe**, will be returned to the Control Shell. In the design file, types of analyses and their parameters can be specified. If the parameters need changes, or analyses need to be added or subtracted, use the **Analysis** menu.

To change the AC analysis parameters hit **A** or move the cursor to **AC & Noise** and hit **Return**. A window will appear that contains a list of parameters and their values. A **yes** or **no** response to **Enable?** tells the computer whether you want an analysis performed. Edit the parameters by hitting **Return** until the cursor

has reached the desired field and then enter the correct values. When all of the parameters have been either accepted or corrected, the question **Done?** will appear. Answer with **Y** or **N**. The same procedure is used for **DC Sweep**, **Transient**, and **Monte Carlo**. **Change Options** specifies the control options for the simulation. After going through the set of options the user is automatically returned to the menu. **Run Temp** is a little different in format. After selecting **Run Temp**, a window appears with one or more numbers on it. To have the analyses run at different temperatures, fill in the desired temperatures in this window and hit **ESC** when done.

The next main menu is **Display**. In the current version of PSpice (4.02), this menu contains only one item,

Print

Print brings up a window that asks the user to enter the output variables desired in the hard copy output. The user must select from among the different analyses by typing the first letter of the chosen analysis and hitting **Return**. Following this, another window will appear that lists the current values to be printed. The list may be added to, edited, or deleted from. Hit **ESC** when done.

The next main menu is **Probe**. **Probe** allows the following selection:

View
All/Some/None
Command File
Log File
Format
Setup

View simply runs **Probe**: hit **V** or move the cursor to **View** and hit **Return**. The operation of **Probe** will be discussed later in this report.

All/Some/None lets the user select which **Probe** variables to save under the current file name with the extension **.dat** or **.txt**. After selecting **All/Some/None**, the question **Probe Variables?** appears. It can be answered with **A** for all variables, **S** for some variables, or **N** for no variables. If **S** is selected then the variables desired must be entered. Hit **ESC** to return to the menu.

Command File allows the user to select whether a command file is used and if so, which one. Select **Command File** in the usual manner and hit **Y** or **N** to choose whether to use one. If one is desired, the computer will request the user to

enter the file's name. The default name is the current file name with the extension **.cmd**.

Log to File allows the user to select whether a log file should be generated. Select **Log to File** in the usual manner and hit **Y** or **N** to choose whether to generate one. If **Y** is selected, the computer will ask the user to enter the file's name. The default name is the current file name with the extension **.cmd**.

Format lets the user select between binary and ASCII intermediate file format. Select **Format** in the usual manner and then type **B** for binary or **T** for ASCII text and hit **Return**.

Setup allows the user to select the display and printer setup information. **Setup** is selected in the usual manner and a window appears asking for the display, the printer port, and the printer type. See Reference 1, starting on page 209, for a list of options. Enter the correct information and hit **Return**. This menu updates the **probe.dev** file.

The last main menu is the **Quit** menu, which contains only two items:

Exit to DOS
DOS Command

Exit to DOS does exactly what it says: it exits the program. If there are changes to the circuit file it will prompt the user to save the changes or discard them.

The **DOS command** will exit the program temporarily and take the user to the prompt. The user is then allowed to execute one **DOS** command. After the command has been executed the user will be returned to the Control Panel.

USING PROBE

Probe is used to view waveform results generated from PSpice. To use **Probe**, the command

.PROBE

must be in the circuit file. This command generates a file, **PROBE.DAT**, that contains the simulation results. After running the PSpice simulation, the computer will automatically enter **Probe**. To run **Probe**, at the prompt give the command

PROBE

It is also possible to specify command, device, log, and display files from the command line. See Reference 1, starting on page 205, for more information. After the command is given, a window opens that contains a menu with four to seven options. The options are:

Exit_program
DC_sweep
AC_sweep
Transient_analysis
All_<type-of-analysis>
Next_temperature_or_circuit
Previous_temperature_or_circuit

Exit_program will return the user to DOS without changing the data file. Just hit **E** or move the cursor to **Exit_program** and hit **Return**.

By pressing the **D** key or moving the cursor to **DC_sweep** and hitting **Return**, the **DC sweep** section of the data file is selected for viewing. A new window appears with a menu at the bottom of the screen and an empty plot at the top of the screen with the words **All voltages and currents are available**. The menu contains the following items:

Exit
Add_trace
X_axis
Y_axis
Plot_control
Display_control
Hard_copy

The menu items are selected in the usual **Probe** format.

Exit will exit the user from the DC analysis.

Add_trace will prompt the user for the node names of the voltages and currents to be plotted. Hit **Return** after entering the voltage and/or the current, and a plot will appear. The user can also perform algebraic functions such as $V(5)*V(4)$ on voltage and current nodes and display the results. After adding a trace, a new menu will also appear with the following items:

Exit
Add_trace
Remove_trace
X_axis
Y_axis
Plot_control
Display_control
Hard_copy
Cursor

If **Add_trace** is selected again, another trace will be added to the same graph. If **Remove_trace** is selected, the user can delete one of the traces. **X_axis** allows the user to select the scale of the x-axis of the plot; either a logarithmic or linear scale may be chosen (the linear scale is the default). **Y_axis** allows the user to select the scale of the y-axis. **Plot_control** allows the user to add another separate plot with its own axes to the screen, or when more than one plot is on the screen, it allows the user to select the desired plot. **Display_control** lets the user save and/or restore screen attributes. **Hard_copy** allows the printing of plots in a 1-page, 2-page, or user-specified format. **Cursor** enables the user to get an exact reading on a position on the plot. It adds crosshairs to the plot and allows the user to move them using the arrow keys. A box in the bottom right-hand corner of the screen reveals the exact numerical location of the crosshairs.

After viewing all that is desired with the DC analysis, the user must return to the main menu to choose another analysis. **AC_sweep** and **Transient_analysis** are run the same way and have the same menus as **DC_sweep**, so they will not be discussed further.

All_<type-of-analysis> is only displayed when there is more than one run of any analysis. It allows the user to display more than one run at a time.

Next_temperature_or_circuit is only displayed if more than one circuit or temperature is in the data file. This option allows the user to select the next temperature or the next circuit if the last temperature or circuit has not been reached.

The **Previous_temperature_or_circuit** option is only displayed if there is more than one temperature and the user is not on the first temperature. This option allows the user to go back to the previous temperature or previous circuit.

Probe is fairly easy to use. It just takes time to get used to the different menu levels. This has been a rather brief outline of its use. Only the most commonly used

functions were described in detail. More information on **Probe** can be found in Reference 1, starting on page 205.

THE STIMULUS EDITOR (StmEd)

Information on the **Stimulus Editor** is found in Reference 1, starting on page 193. **StmEd** was not used on the project documented here. **StmEd** allows the user to make changes to the transient input. In this report, changes to the inputs were made directly into the circuit file. For visualization purposes however, it might be easier to use **StmEd**. It should be fairly easy to use as it looks similar to **Probe**.

CAPABILITIES OF PSPICE AND MICRO-CAP III

PSpice and MICRO-CAP III have many of the same capabilities. If a certain type of solution is required, it is usually possible to obtain the results from either program. The solutions might come in different formats or be easier or more difficult to obtain, but they are available.

COMPARISON OF THE TWO PROGRAMS

In PSpice, the circuit is put into a file by assigning node numbers to the different parts. MICRO-CAP III has a schematic capture package, which means that the circuit can be drawn and visualized instead of merely seen as a list of node numbers. Compiling a list of node numbers is not difficult: the circuit must be drawn freehand and a node number assigned at each junction. As long as care is taken, this is a rather simple process. Changes to the circuit add complications. Changes can be made using either program. However, it is easier to grab the component on screen and add it to the circuit than to reassign node numbers and verify that all node numbers are updated on the file. Therefore, both programs make it fairly easy to enter a circuit, but MICRO-CAP III makes it a little easier to visualize the circuit and to make changes to it.

Both PSpice and MICRO-CAP III can perform a DC analysis. The output information, however, is given in slightly different formats. To obtain a node voltage with MICRO-CAP III, a plot is run (and a table if requested) of the node voltage with respect to the input voltage range. A sample plot is shown in Figure 2. If several different node voltages are needed, a plot must be run every time. PSpice

calculates a table that lists all the voltages at the specific operating point. With PSpice an unlimited number of node voltages can be requested per single run. Therefore, it is not necessary to run a simulation every time a node voltage is needed.

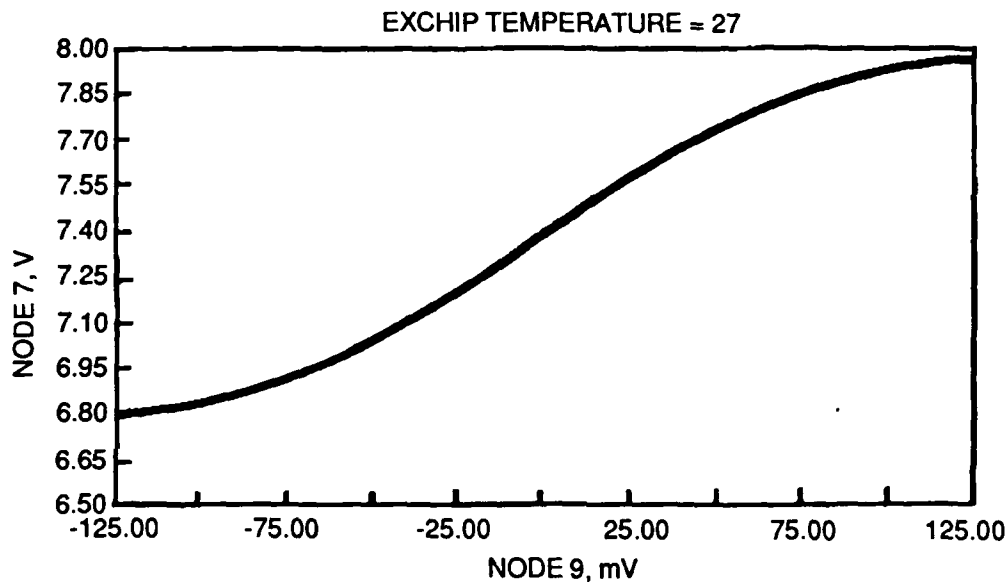


FIGURE 2. MICRO-CAP III DC Plot.

The outputs obtained from the AC analyses from both programs are similar. Tables and graphs are output from both programs. The main objective for an AC analysis, at least in the work done for this report, is to obtain a logarithmic graph, seen in Figure 3, that will show the gain and the 3 dB frequency. Results from a lot of the nodes are not needed. In this case the information is just as easy to extract from both programs.

The Transient analysis output requirements (most importantly, a graph) are similar to the AC analysis requirements. A graph of the output voltage with respect to time is shown in Figure 4. The circuit rise time can be calculated from this graph. Because both PSpice and MICRO-CAP III can output this information in the same format, there is no preferred program in this case.

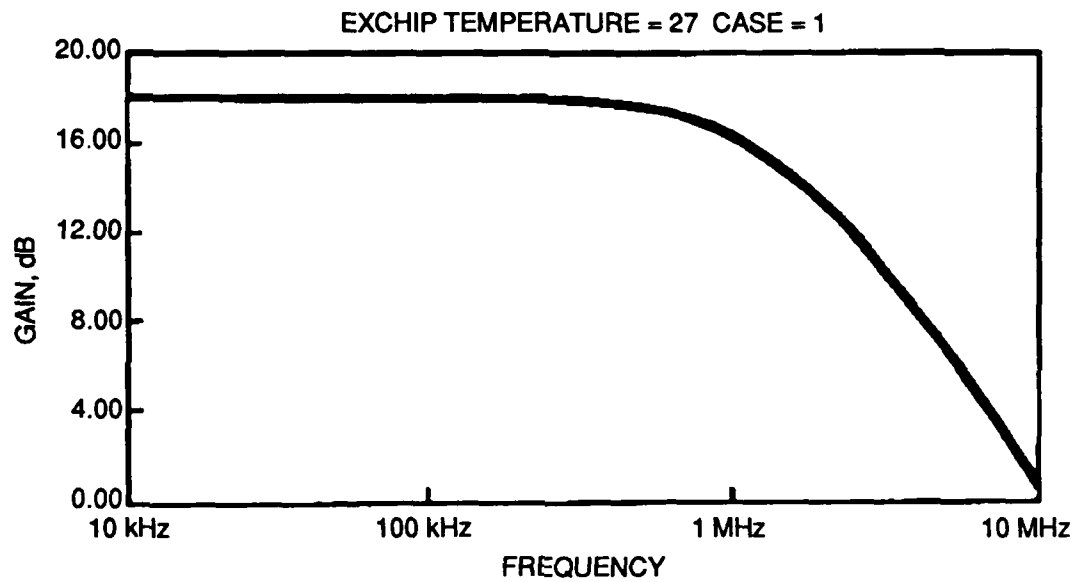


FIGURE 3. MICRO-CAP III AC Plot.

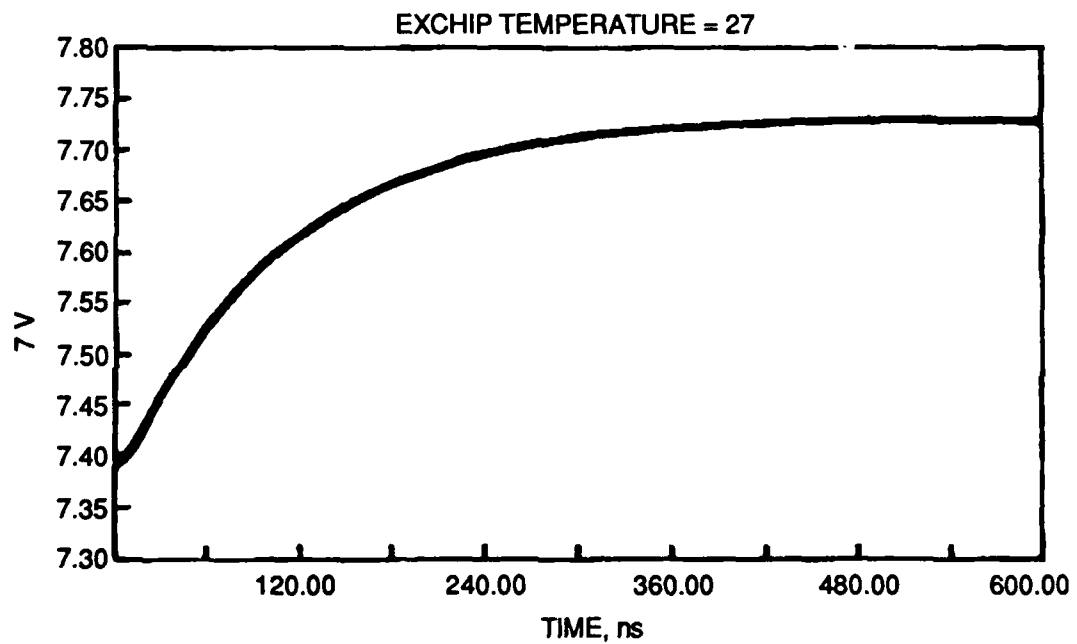


FIGURE 4. MICRO-CAP III Transient Plot.

A **Monte Carlo** analysis can be performed with either PSpice or MICRO-CAP III. Three distributions are available with MICRO-CAP III: Gaussian, linear, and worst case. PSpice has these three distributions available and also allows the user to specify his own. With PSpice, a **Monte Carlo** analysis can be performed while varying the value of any device or device parameter. With MICRO-CAP III, the analysis can only be performed when varying a device parameter. The only way to vary the value of a resistor, capacitor, etc., is to step the component value, which does not produce a true distribution. Numerical output can be obtained from both programs. With PSpice, however, the user is able to specify which output and how many runs are wanted. With MICRO-CAP III, all the output from all the runs is generated. MICRO-CAP III also produces a nice graph that overlays all the graphs from all the cases onto one plot, graphically showing the user the range of values that could be encountered. PSpice will draw graphs for each case, but on separate plots in the output file. In **Probe**, the **All_<type of analysis>** option allows the user to view all cases on a single plot. Figure 5 is an example of a MICRO-CAP III **Monte Carlo** graph.

As mentioned above, MICRO-CAP III allows the values of components (such as resistors) to be stepped. In other words, the resistor value could be varied from 1 k Ω to 5 k Ω in 500 Ω steps. PSpice can only perform a **Monte Carlo** analysis on these devices. Temperature stepping is possible with both programs. With PSpice, however, every temperature needed for the analyses must be specified, and every analysis that is requested in the circuit file is run with each temperature. With MICRO-CAP III, the temperature is actually stepped like the resistor used in the example above. The range and step size are specified. Only the analysis currently being requested will be run with the temperatures. In this case, MICRO-CAP III seems easier to use.

A Fourier analysis can be run with either PSpice or MICRO-CAP III. PSpice does not output a plot and only gives the values of the fundamental and the first eight harmonics in the output file. The results for as many harmonics as desired, however, can be obtained from **Probe**, PSpice's graphics post-processor. **Probe** produces no tabular output, but has cross hairs that can be moved to obtain the exact value of a specific point on a plot. MICRO-CAP III seems to be easier to use in this case also. MICRO-CAP III lets the user specify the number of harmonics from a menu and then outputs plots of the magnitude, angle, sine, and cosine. Tabular output is also available. A sample output screen is shown in Figure 6.

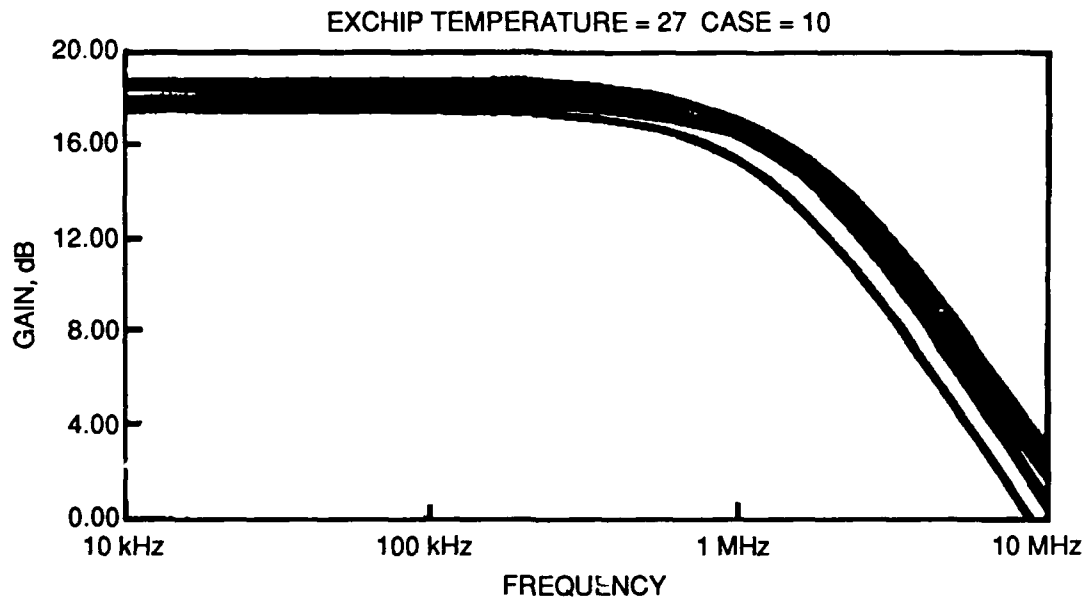


FIGURE 5. MICRO-CAP III AC Monte Carlo Plot.

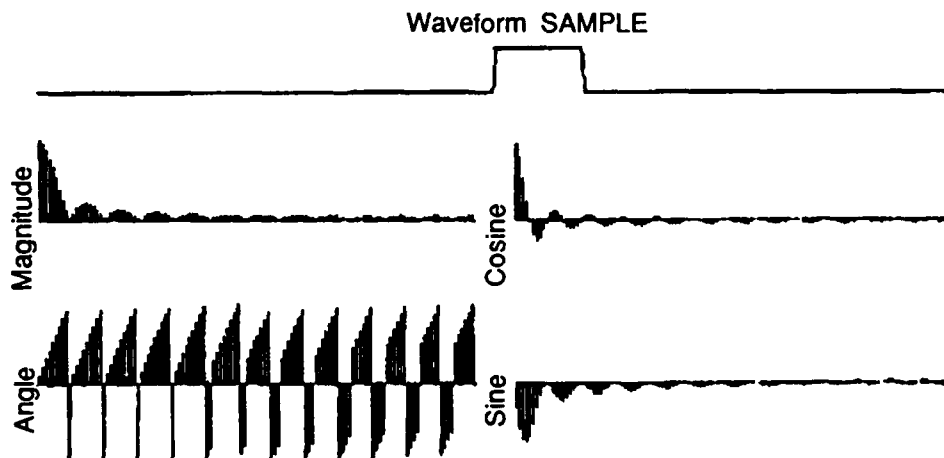


FIGURE 6. MICRO-CAP III Fourier Plots.

A noise analysis can be run in either PSpice or MICRO-CAP III. With MICRO-CAP III the noise analysis is run along with the AC analysis as an output option. The output can be graphical and/or tabular. PSpice has a separate command for the noise analysis and this output can also be graphical and/or tabular.

One last thing to mention about the capabilities of PSpice compared to MICRO-CAP III is that the output tables from PSpice are much easier to print than those from MICRO-CAP III. Because the MICRO-CAP III tables are usually larger than the screen size, they cannot be printed from the screen. It is necessary to store them in a file, back out of MICRO-CAP III, and then print them in DOS, which can be cumbersome.

EXAMPLE CIRCUITS

DIFFERENTIAL AMPLIFIER

The first circuit that was built and tested was a basic differential amplifier, as shown in Figure 7. The circuit contained four 2N2222A transistors, several 1% resistors, and a capacitor. The two DC voltage supplies were +12 and -12 volts. V_{IN} was the small signal input.

The circuit was DC analyzed best by splitting the circuit into two parts as shown in Figure 8. To simplify the analysis, it was assumed that the β s of the transistors were large and the following equation was used

$$I_T = (V_{CC} - V_{BE} - V_{EE}) / R_{bias} \quad (2)$$

Because $V_{CC} = 12$ volts, $V_{BE} = 0.7$ volt, $V_{EE} = -12$ volts, and $R_{bias} = 20 \text{ k}\Omega$, then $I_T = 1.17 \text{ mA}$. $I_C = 582.5 \text{ }\mu\text{A}$ since

$$I_C = I_T / 2$$

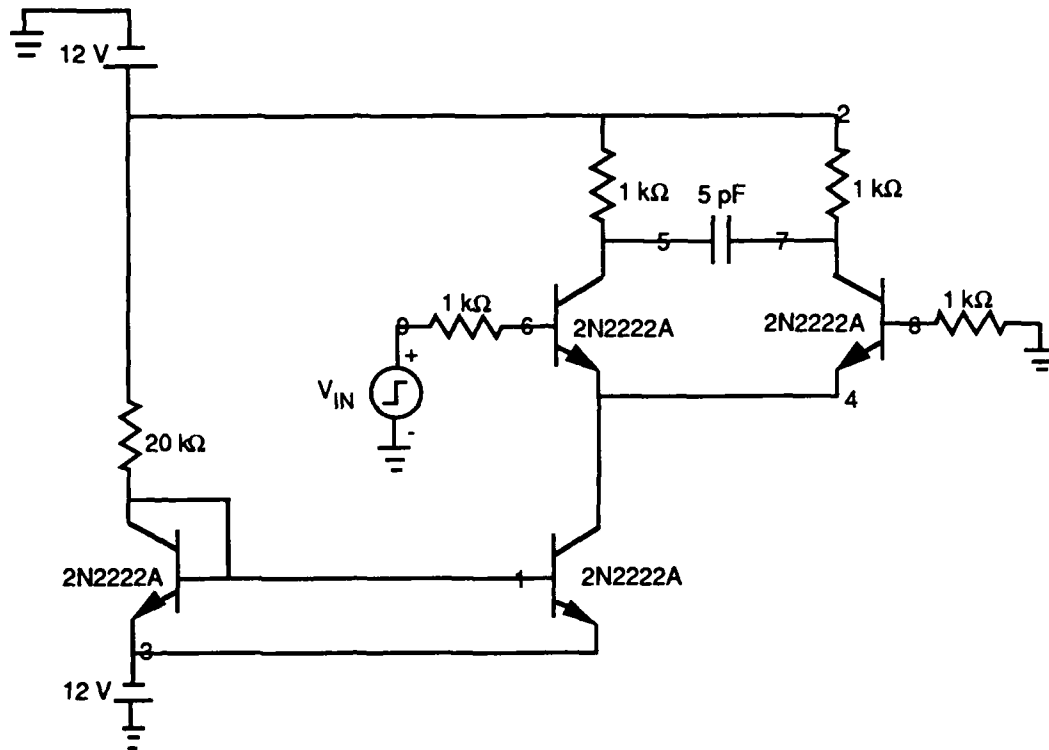


FIGURE 7. Differential Amplifier Circuit Built With Individual Transistors.

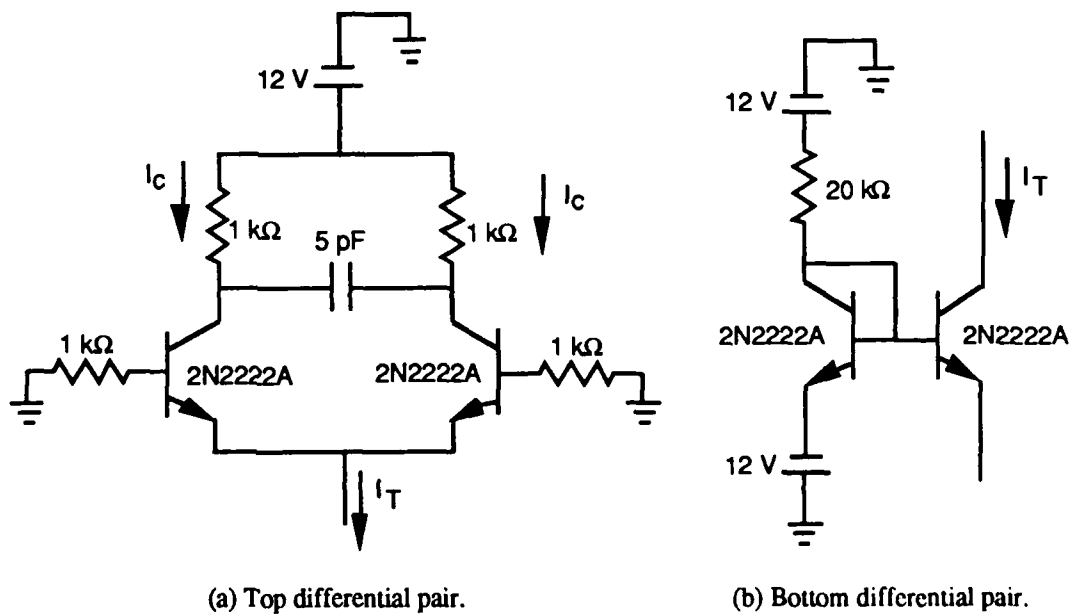


FIGURE 8. Differential Amplifier (From Figure 7) Split in Two Parts for DC Analysis.

The collector voltage was calculated from

$$V_C = V_{CC} - I_C R_C$$

If $R_C = 1\text{ k}$, then $V_C = 11.42\text{ volts}$ and if $R_C = 10\text{ k}$, then $V_C = 6.18\text{ volts}$. The circuit with its voltages and currents is shown in Figure 9. The values in parentheses are the values when using $R_C = 10\text{ k}$. All other values refer to $R_C = 1\text{ k}$. Table 2 compares the DC results of the circuit. The results calculated by hand, in the lab, with PSpice, and with MICRO-CAP III are included in the table. Appendix A contains a block diagram of the test setup, Appendix B contains a listing of the PSpice input file, and Appendix C contains the MICRO-CAP III circuit diagram.

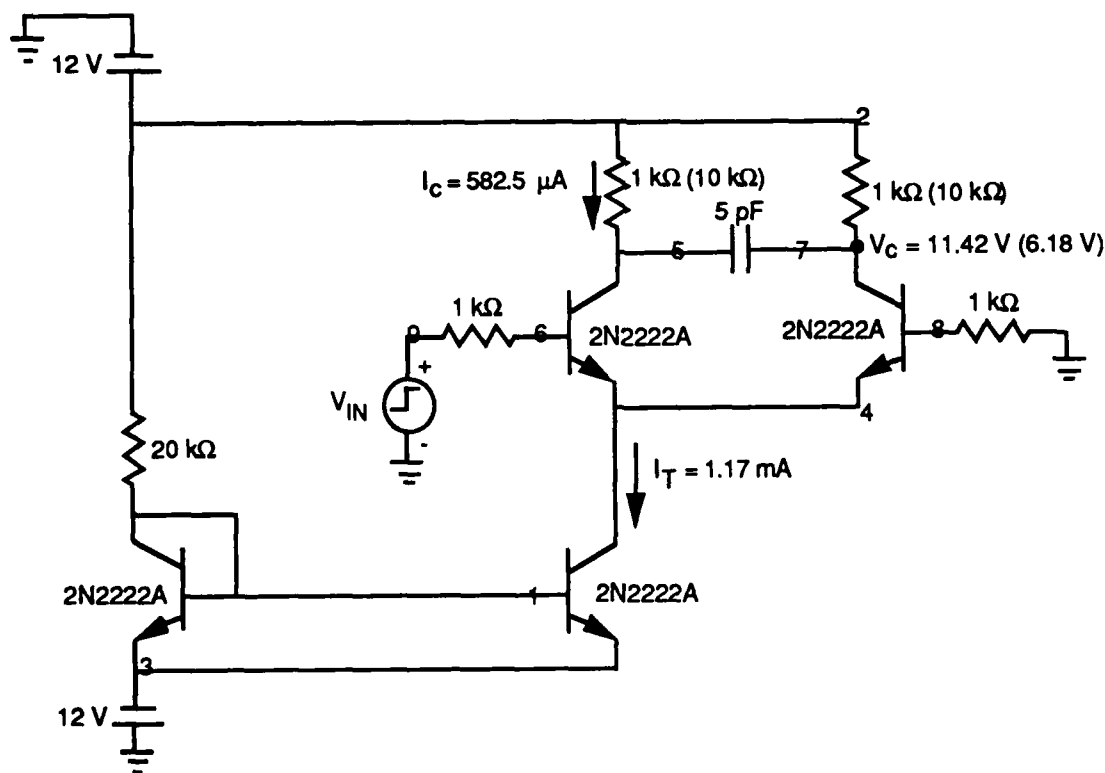


FIGURE 9. Differential Amplifier Solution.

TABLE 2. Differential Amplifier DC Results (Individual Transistors, Figure 9).

| RC | Node no. | | Voltage, volts | | | |
|-----|----------|-----|----------------|--------|--------|--------|
| | PSpice | MC3 | PSpice | MC3 | Lab | Calc |
| 1k | 5 | 7 | 11.34 | 11.41 | 11.57 | 11.42 |
| | 4 | 5 | 11.34 | 11.41 | 11.59 | 11.42 |
| | 6 | 4 | -0.64 | -0.64 | -0.61 | -0.70 |
| | 7 | 1 | -11.35 | -11.35 | -11.37 | -11.30 |
| 10k | 5 | 7 | 5.45 | 6.2 | ~8.41 | 6.18 |
| | 4 | 5 | 5.45 | 6.2 | ~8.10 | 6.18 |
| | 6 | 4 | -0.64 | -0.64 | -0.62 | -0.70 |
| | 7 | 1 | -11.35 | -11.35 | -11.36 | -11.30 |

To perform an AC analysis on the circuit, Miller's theorem was used. The following is a brief review of this Theorem. Figure 10 is an example circuit that is used to explain Miller's theorem.

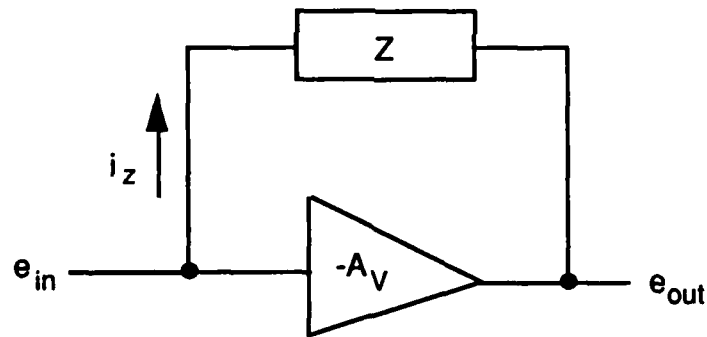


FIGURE 10. Example Circuit to Demonstrate Miller's Theorem.

From Figure 10, the following equation can be derived.

$$e_{out} = -A_v e_{in} \quad (3)$$

The current, i_z , can be calculated with

$$i_z = (e_{in} - e_{out}) / z \quad (4)$$

Substituting Equation 3 into Equation 4 gives

$$i_z = [e_{in}(1 + A_v)] / z \quad (5)$$

Therefore the input impedance, Z_{in} , due to z is

$$Z_{in} = e_{in} / i_z = z / (1 + A_v) \quad (6)$$

To calculate the output impedance, Z_{out} , the following equation is used for i_z .

$$i_z = (e_{in} - e_{out}) / z \quad (7)$$

Substituting Equation 3 into Equation 7 gives

$$i_z = -[e_{out}(1 / A_v + 1)] / z \quad (8)$$

Therefore the output impedance is

$$Z_{out} = e_{out} / (-i_z) = z / (1 + 1 / A_v) \quad (9)$$

Figure 10 therefore converts to Figure 11 . Applying Miller's theorem to Figure 12a gives the equivalent circuit shown in Figure 12b . Converting Figure 12a to a more generalized circuit gives Figure 13a. Figure 13a can be converted to Figure 13b by again using Miller's theorem. Therefore, the drop across z is $2e_o$. The equation for i_z is

$$i_z = 2e_o / z \quad (10)$$

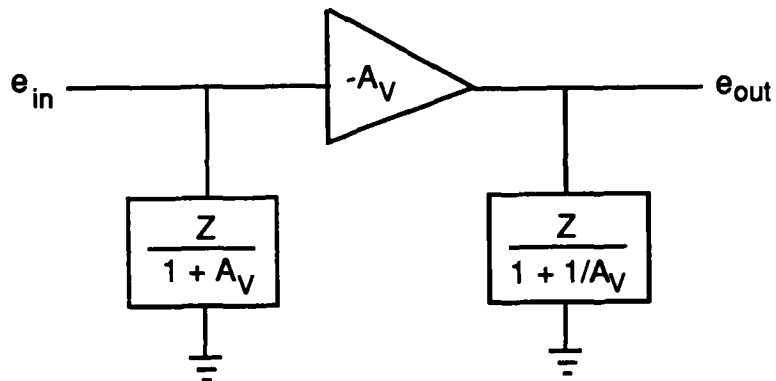


FIGURE 11. Example Circuit From Figure 10 Converted Using Miller's theorem.

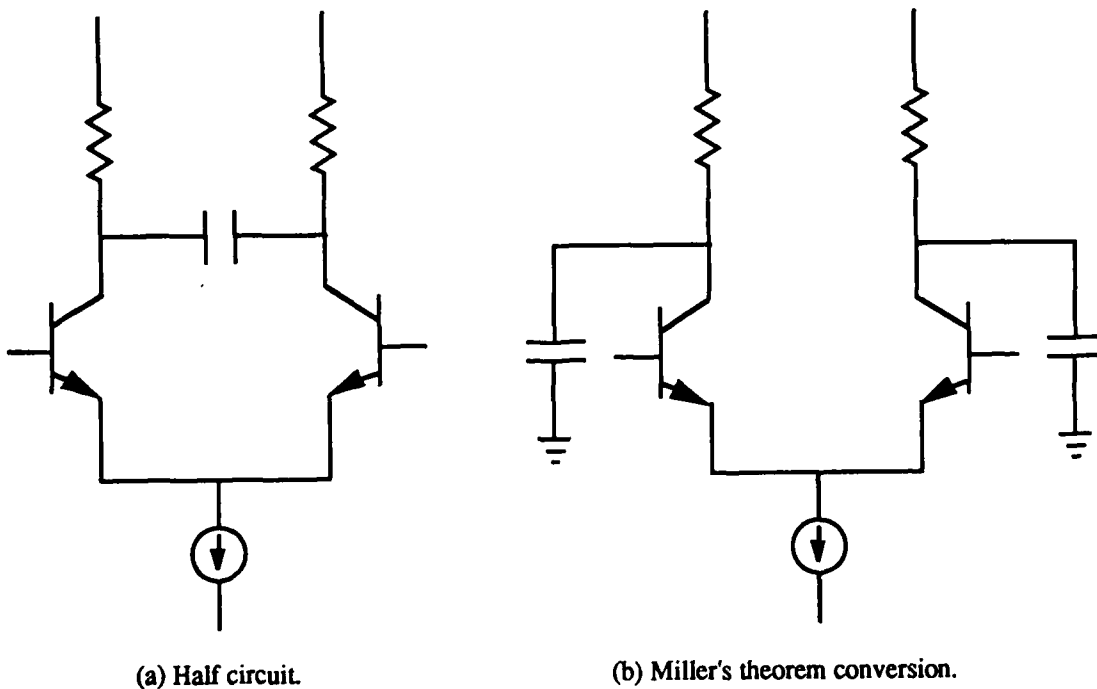


FIGURE 12. Differential Amplifier Half Circuit Converted Using Miller's Theorem.

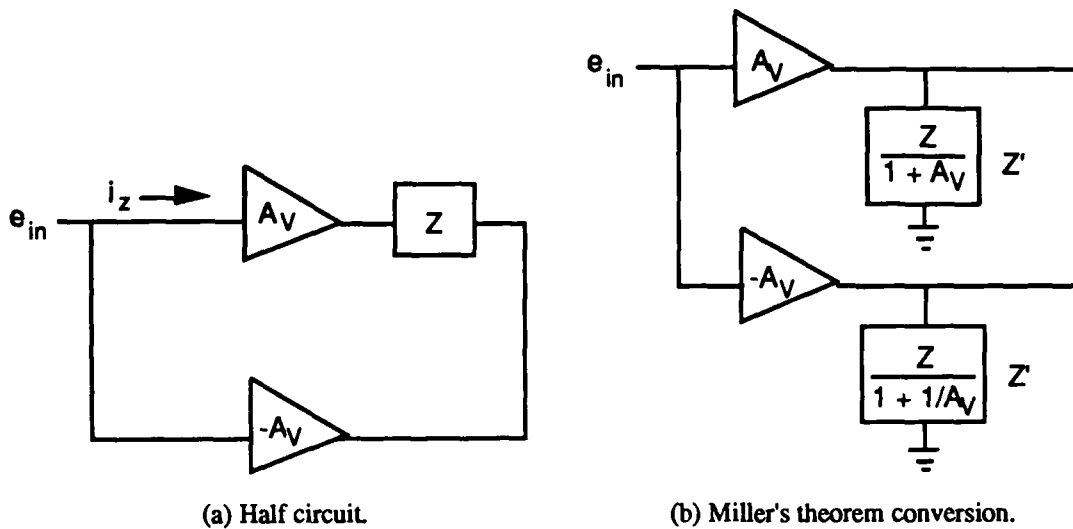


FIGURE 13. Generalized Form of Differential Amplifier Half Circuit Converted Using Miller's Theorem.

And if z' is used to represent $z/(1+A_V)$ then

$$z' = e_o / i_z \quad (11)$$

Combining Equations 10 and 11 gives

$$z' = z / 2 \quad (12)$$

Since

$$z' = 1 / (j2\pi fC) \quad (13)$$

then

$$z' = 1 / [(j2\pi f)(2C)] \quad (14)$$

and the value of the z' (capacitors in Figure 12b) is $2C$. Figure 12b can be converted into a simple RC circuit as shown in Figure 14 and the 3 dB frequency of the circuit is

$$f_{3dB} = 1 / [(2\pi R_c)(2C)] \quad (15)$$

which gives

$$f_{3dB} = 1 / (4\pi R_c C) \quad (16)$$

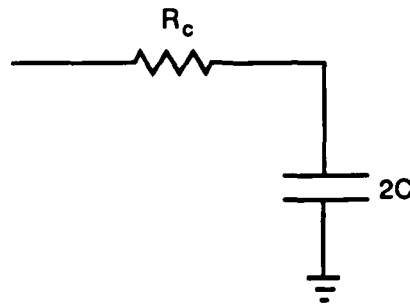


FIGURE 14. RC Circuit.

Substituting the circuit values into Equation 16 gives the values shown in Table 3. The 3 dB frequency point of the output was also found in the lab with the setup shown in Appendix A. This frequency point was calculated by using a low voltage (approximately 30 mV) sine wave signal for V_{IN} and then checking the magnitude of the output signal at 1 kHz on the oscilloscope. The output signal was then multiplied by 0.707 to give the magnitude of the 3 dB signal. The frequency of V_{IN} was decreased until it reached the 0.707 level and that frequency was recorded as the 3 dB frequency. With MICRO-CAP III and PSpice, gain versus frequency plots were charted to reveal the 3 dB frequency. These plots also revealed the circuit gain, as shown in Table 4. Circuit gain in the lab was calculated by taking the output voltage at 1 kHz and dividing it by the input voltage. By hand, the low frequency gain of the circuit was calculated using the following equation:

$$i_c = I_s e^{V_{BE}/V_T} \quad (17)$$

which is explained in Reference 3, page 402. This equation is expanded to

$$i_{c1} = I_s e^{(V_{IN} - V_E)/V_T} \quad (18)$$

$$i_{c2} = I_s e^{-V_E/V_T} \quad (19)$$

TABLE 3. Differential Amplifier AC Results (Individual Transistors, Figure 9).

| R_C | C | f_{3dB} | | | |
|---------------|--------|-----------|---------|----------|----------|
| | | PSpice | MC3 | Lab | Calc |
| 1 k Ω | 5 pF | 1.26 MHz | 1.5 MHz | 1.25 MHz | 15.9 MHz |
| 1 k Ω | 300 pF | 251 kHz | 220 kHz | 210 kHz | 265 kHz |
| 10 k Ω | 300 pF | 25.1 kHz | 20 kHz | 24.5 kHz | 26.5 kHz |

TABLE 4. Differential Amplifier Gain Results (Individual Transistors, Figure 9).

| R | C | Gain, dB | | |
|---------------|--------|----------|-------|----------|
| | | PSpice | MC3 | Lab/Calc |
| 1 k Ω | 300 pF | 20.81 | 19.78 | 23.08 |
| 1 k Ω | 5 pF | 20.81 | 19.78 | 22.77 |
| 10 k Ω | 300 pF | 40.11 | 39.78 | 42.77 |

Combining Equations 18 and 19 gives

$$i_{c1} / (i_{c1} + i_{c2}) = (e^{V_{IN}/V_T}) / (e^{V_{IN}/V_T} + 1) \quad (20)$$

and

$$i_{c2} / (i_{c1} + i_{c2}) = (e^{-V_{IN}/V_T}) / (1 + e^{-V_{IN}/V_T}) \quad (21)$$

Rearranging the equations gives

$$i_{c1} / (i_{c1} + i_{c2}) = 1 / (1 + e^{-V_{IN}/V_T}) \quad (22)$$

and

$$i_{c2} / (i_{c1} + i_{c2}) = 1 / (e^{V_{IN}/V_T} + 1) \quad (23)$$

Because

$$i_{c1} + i_{c2} = I_T \quad (24)$$

then

$$i_{c1} = I_T / (1 + e^{-V_{IN}/V_T}) \quad (25)$$

and

$$i_{c2} = I_T / (1 + e^{V_{IN}/V_T}) \quad (26)$$

The equation for the output voltage is then

$$V_c = V_{cc} - I_T R_C / (1 + e^{V_{IN}/V_T}) \quad (27)$$

When performing a small signal analysis, the DC voltage is taken out. The DC voltage is

$$V_{DC} = V_{cc} - I_T R_C / 2 \quad (28)$$

Subtracting V_{DC} from V_c leaves

$$V_o = -I_T R_C / (1 + e^{V_{IN}/V_T}) + I_T R_C / 2 \quad (29)$$

Rearranging the above equation

$$V_o = I_T R_C \left[1/2 - 1 / (1 + e^{V_{IN}/V_T}) \right] \quad (30)$$

or

$$V_o = (1/2) I_T R_C \left[(e^{V_{IN}/V_T} - 1) / (e^{V_{IN}/V_T} + 1) \right] \quad (31)$$

From Reference 4, page 168, it can be seen that

$$\text{Tanh}[V_{IN}/(2V_T)] = (e^{V_{IN}/V_T} - 1)/(e^{V_{IN}/V_T} + 1) \quad (32)$$

Therefore

$$V_o = (1/2)I_T R_c \text{Tanh}(V_{IN}/2V_T) \quad (33)$$

If V_{IN} is small then

$$V_o = (1/2)I_T R_c (V_{IN}/2V_T) \quad (34)$$

The gain of the circuit is then

$$V_o/V_{IN} = I_T R_c / 4V_T \quad (35)$$

The rise time, or the time constant, τ , is found by doing a Transient analysis. τ is found with PSpice and MICRO-CAP III by making V_{IN} a low frequency (1 kHz), small width (10 μ s), small magnitude (50 mV) pulse. The rise time is the time that it takes the output to go from 10% of its peak value to 90% of its peak value. The same calculation was made in the lab on the oscilloscope. To calculate the time constant by hand

$$\tau = 0.35 / f_{3dB} \quad (36)$$

A comparison of the values found for τ can be seen in Table 5. As mentioned earlier, if V_{IN} is very small (much less than 5 mV), then

$$V_o/V_{IN} = I_T R_c / 4V_T \quad (37)$$

If V_{IN} is larger, then the gain can be found by comparing the magnitude of the output waveform to the input pulse. Using Figure 15, the gain can be calculated using the following equation:

$$V_o/V_{IN} = (B - A)/C \quad (38)$$

A comparison of the Transient results, using Equation 38, can be seen in Table 5.

TABLE 5. Differential Amplifier Transient Results
(Individual Transistors, Figure 9).

| C | τ | | | |
|--------|--------------|-------------|-------------|--------------|
| | PSpice | MC3 | Lab | Calc |
| 300 pF | 1.25 μ s | 1.4 μ s | 1.4 μ s | 1.32 μ s |
| 5 pF | 180 ns | 200 ns | 280 ns | 22 ns |

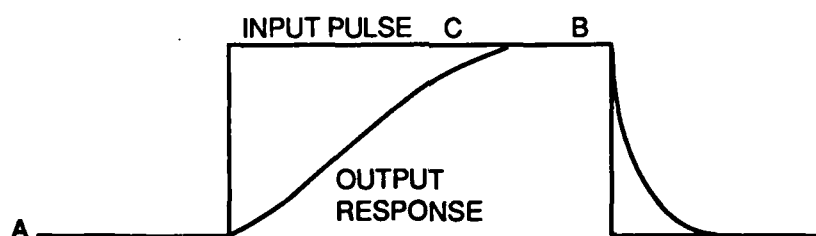


FIGURE 15. Transient Analysis Input Pulse and Output Response.

Because the lab values were slightly different from those of either PSpice or MICRO-CAP III, the same circuit was built with transistors found on a chip. The chip was an Interdesign MO-001, which contained four NPN transistors requiring 8 V and -8 V supply voltages. The transistors were not 2N2222As. In fact, as the results showed, they worked a little better. Therefore, the results for the circuit built with the chip compared to what PSpice and MICRO-CAP III calculated, were different. The results with the circuit built with the chip are found in Tables 6 through 9.

TABLE 6. Differential Amplifier DC Results (Chip Transistors, Figure C-2).

| Node no. | | Voltage, V | | | |
|----------|-----|------------|-------|-------|-------|
| PSpice | MC3 | PSpice | MC3 | Lab | Calc |
| 5 | 7 | 7.37 | 7.41 | 7.38 | 7.41 |
| 4 | 5 | 7.37 | 7.41 | 7.38 | 7.41 |
| 6 | 4 | -0.64 | -0.64 | -0.72 | -0.70 |
| 7 | 1 | -7.35 | -7.35 | -7.31 | -7.30 |

NWC TP 7075

TABLE 7. Differential Amplifier AC Results
(Chip Transistors, Figure C-2).

| C | f_{3dB} | | | |
|--------|-----------|---------|---------|----------|
| | PSpice | MC3 | Lab | Calc |
| 5 pF | 1.3 MHz | 1.1 MHz | 4.2 MHz | 15.9 MHz |
| 300 pF | 230 kHz | 230 kHz | 250 kHz | 265 kHz |

TABLE 8. Differential Amplifier Gain Results
(Chip Transistors, Figure C-2).

| C | Gain, dB | | |
|--------|----------|-------|----------|
| | PSpice | MC3 | Lab/Calc |
| 5 pF | 20.48 | 19.83 | 20.79 |
| 300 pF | 20.47 | 19.82 | 20.52 |

TABLE 9. Differential Amplifier Transient Results
(Chip Transistors, Figure C-2).

| C | τ | | | |
|--------|-------------|-------------|--------------|--------------|
| | PSpice | MC3 | Lab | Calc |
| 5 pF | 205 ns | 260 ns | 80 ns | 22 ns |
| 300 pF | 1.5 μ s | 1.5 μ s | 1.33 μ s | 1.32 μ s |

VIDEO AMPLIFIER

The second circuit chosen to test was the video amplifier shown in Figure 16. First, the DC voltages and currents were calculated by hand to compare to the computer and lab results. To calculate the current through R_{BB} the following node equation was used

$$V_{EE} = V_{CC} - I(R_{BB}) - V_{BE} - I(R_E) \quad (39)$$

or

$$8 - I(7.15 \text{ K}) - 0.7 - I(1 \text{ K}) = -8$$

Solving the above equation

$$I = (V_{CC} - V_{EE} - V_{BE}) / (R_{BB} + R_E) = 1.88 \text{ mA}$$

with

$$V(3) = V_{CC} - I(R_{BB}) = 8 - (1.88 \text{ mA})(7.15 \text{ K}) = -5.44 \text{ V}$$

Subtracting V_{BE} to calculate $V(11)$

$$V(11) = V(3) - V_{BE} = 5.44 - 0.7 = -6.14 \text{ V}$$

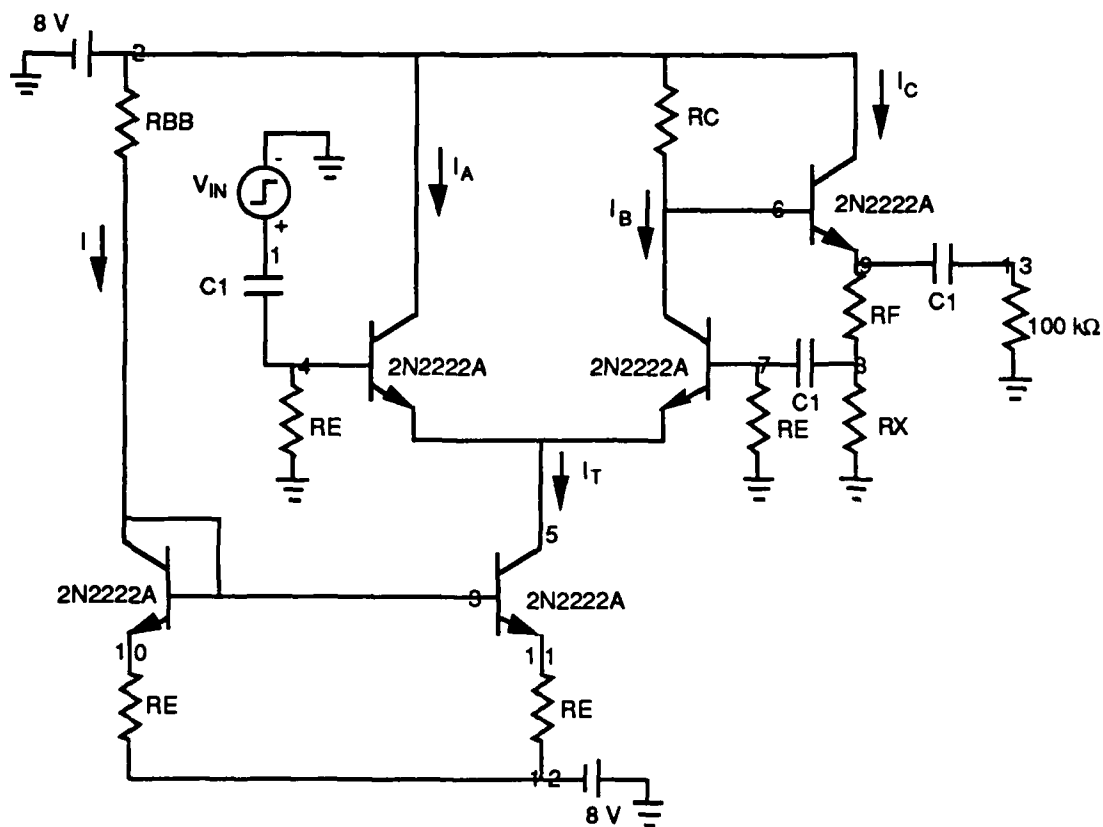


FIGURE 16. Video Amplifier Circuit.

It is found by

$$I_T = [V(11) - V_{EE}] / R_E = (-6.14 + 8) / 1 \text{ K} = 1.8 \text{ mA}$$

NWC TP 7075

Because

$$V(4) = V(7) = 0$$

Therefore

$$V(5) = V(4) - 0.7 = 0 - 0.7 = -0.7 \text{ V}$$

Because of symmetry

$$I_A = I_B = I_T / 2 = 0.93 \text{ mA}$$

Solving for V(6)

$$V(6) = V_{CC} - I_B R_C = 8 - (0.93 \text{ mA})(3.01 \text{ K}) = 5.2 \text{ V}$$

and for V(9)

$$V(9) = V(6) - V_{BE} = 5.2 - 0.7 = 4.5 \text{ V}$$

Therefore, I_C can be calculated

$$I_C = V(9) / (R_F + R_X) = 4.5 / 5 \text{ K} = 0.9 \text{ mA}$$

and V(8)

$$V(8) = I_C R_X = (0.9 \text{ m})(0.5 \text{ K}) = 0.45 \text{ V}$$

Table 10 is a comparison of these values to those found in the lab and those calculated on the computer.

TABLE 10. Video Amplifier DC Results (C = Chip Transistors, I = Individual Transistors, Figure 16).

| Node no. | Voltage, V | | | | |
|----------|------------|-------|-------|-------|-------|
| | PSpice | MC3 | Lab-C | Lab-I | Calc |
| 3 | -5.47 | -5.46 | -5.38 | -5.48 | -5.44 |
| 10 | -6.13 | -6.13 | -6.14 | -6.12 | -6.14 |
| 5 | -0.65 | -0.65 | -0.75 | -0.64 | -0.70 |
| 6 | 5.24 | 5.21 | 5.29 | 4.82 | 5.20 |
| 8 | 0.46 | 0.47 | 0.46 | 0.42 | 0.45 |
| 9 | 4.60 | 4.56 | 4.55 | 4.20 | 4.50 |

Figure 16 can be redrawn as Figure 17. The open loop gain, A_o , is

$$A_o = I_T R_c / 4V_T \quad (40)$$

and if the circuit is at a temperature of 27°C, V_T can be calculated to be 0.0258 by

$$V_T = KT / q = (1.38 \times 10^{-23})(273 + 27) / (1.602 \times 10^{-19})$$

where K is Boltzman's constant in joules/kelvin, T is temperature in degrees kelvin, and q is the charge of an electron in volts. Because I_T is 1.86 mA and R_c is 3.01 kΩ then

$$A_o = (1.88 \text{ m})(3.01 \text{ K}) / 4(0.0258) = 54.5$$

which is 34 dB. The ideal low frequency closed loop gain is calculated by using Figure 18. I is calculated to be

$$I = V_f / (R_f + R_x) \quad (41)$$

and, therefore, V_x is

$$V_x = IR_x = V_f R_x / (R_f + R_x) \quad (42)$$

which gives an ideal closed loop gain of

$$A_v = V_f / V_x = (R_f + R_x) / R_x \quad (43)$$

If $R_f = 4.5 \text{ k}\Omega$ and $R_x = 0.5 \text{ k}\Omega$ then

$$A_v = (4.5 \text{ K} + 0.5 \text{ K}) / 0.5 \text{ K} = 10 = 20 \text{ dB}$$

The results of the gain calculated by hand, by computer, and in the lab are found in Table 11. The results in the lab and on the computer were calculated in the same manner as in the differential amplifier circuit.

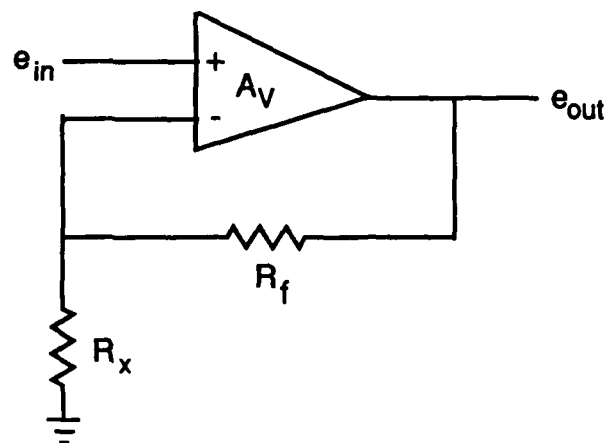


FIGURE 17. Video Amplifier Equivalent Circuit.

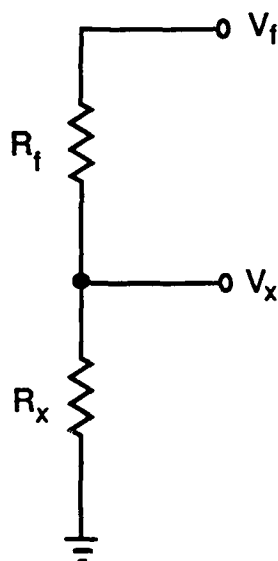


FIGURE 18. Video Amplifier Partial Circuit.

TABLE 11. Video Amplifier Gain Results in Decibels (C = Chip Transistors, I = Individual Transistors, Figure 16).

| PSpice | MC3 | Lab-C | Lab-I |
|--------|-----|-------|-------|
| 21 | 21 | 20.3 | 17 |

The 3 dB frequency point found by performing an AC analysis was not calculated by hand (the equations were too cumbersome). The results found in the lab and on the computer are summarized in Table 12.

The time constant results from the Transient analysis are found in Table 13. This calculation was not made by hand either.

TABLE 12. Video Amplifier AC Results (C = Chip Transistors, I = Individual Transistors, Figure 16).

| Variable | Frequency | | | |
|-----------|-----------|--------|----------|--------|
| | PSpice | MC3 | Lab-C | Lab-I |
| f_{3dB} | 8 MHz | 6 MHz | 19.7 MHz | 7 MHz |
| f_T | 55 MHz | 50 MHz | >50 MHz | 50 MHz |

TABLE 13. Video Amplifier Transient Results (C = Chip Transistors, I = Individual Transistors, Figure 16).

| τ , ns | | | |
|-------------|-----|-------|-------|
| PSpice | MC3 | Lab-C | Lab-I |
| 39 | 52 | 23 | 62 |

Because the results from the lab did not seem to be close enough to the computer results, a circuit using actual 2N2222A transistors was built. The results from that circuit are also given in Tables 10 through 13.

COMPARISON OF RESULTS

In this section, the results of the four circuits built will be compared to the results from PSpice, MICRO-CAP III, and the hand calculations. The comparison tables can be found in the previous section.

DIFFERENTIAL AMPLIFIER (TRANSISTOR)

As shown in Table 2, the DC results from the differential amplifier built with the individual transistors are fairly good for the $R_C = 1 \text{ k}\Omega$ case. For the $R_C = 10 \text{ k}\Omega$ case, the voltages drifted quite a bit. The value recorded in the lab was the value read as soon as voltage was applied to the circuit. This value tended to drift lower through time. The reason for this drifting was assumed to be mismatched transistors: the transistor areas were probably different, causing the I_s 's of the transistors to be different. The transistors had a greater effect when $10 \text{ k}\Omega$ resistors were used, which was proven when the transistors on a chip were used. The DC results improved greatly.

The results for the Transient analysis can be seen in Table 5. The results for the 300 pF capacitor were very close but the results for the 5 pF capacitor were significantly different. The hand calculated value was not even close because when a 5 pF capacitor is used the circuit is essentially transistor dominant. Because the hand calculations take only R_C and C_{LOAD} into account (and not the transistor), the results are off. There is, however, also a large difference between the computer generated rise time and the lab results. The computer models must have been better than the transistor in this area.

The AC results, shown in Table 3, were fairly close except for the hand calculated 5 pF case. This was probably because of the way the calculations were done also, since this circuit was transistor dominant, as in the Transient analysis.

The gains of the circuits are compared in Table 4. In this case, the computer seemed to underestimate the gains by about 3 dB . This is quite a difference, but a designer can plan on getting at least what the computer programs tell him. PSpice seemed to be closer to the correct values than MICRO-CAP III.

DIFFERENTIAL AMPLIFIER (CHIP)

The DC values, shown in Table 6, were very close because this time the differential amplifier circuit was built with a chip containing matched transistors. In this case, MICRO-CAP III and PSpice were excellent simulators.

The Transient analysis results, shown in Table 9, were once again fairly close for the 300 pF capacitor case, but quite different for the 5 pF case. It is understandable that the hand calculated value for the 5 pF case would be off

because of the transistor dominance, but the computer tended to overcalculate the rise time, possibly because the transistors on the chip were not actual 2N2222A transistors. They must have been faster transistors.

The AC analysis results, shown in Table 7, were off by quite a bit. Again, the hand calculated 5 pF case was drastically incorrect because, as mentioned earlier, the circuit was transistor dominant. The equations used to calculate the 3 dB frequency point do not take the transistors into account. Also, the values calculated by the computer were not as good as those found in the lab, probably because the transistors were not actual 2N2222As; they were a better model.

The voltage gain calculations came out fairly close, as seen in Table 8. In this case, PSpice seems to have made the closer calculation to the actual. Remember, however, that the chips used were better than 2N2222As, so possibly MICRO-CAP III was closer to the actual. In the individual transistor circuit discussed earlier, PSpice came closer to the actual value.

VIDEO AMPLIFIER (CHIP)

The DC values, found in Table 10, were once again fairly consistent.

The Transient analysis results, shown in Table 13, were not consistent. The lab values were better than both computer values possibly because the transistors on the chip were better. Unfortunately, there was not much of a correspondence between the computer results.

The AC analysis results, in Table 12, show that once again, because of the superior chip transistors, the 3 dB frequency point found in the lab was much larger than what the computer calculated. The point where the gain reaches 0 dB is symbolized by f_T . The f_T of the circuit was found to be greater than 50 MHz, probably because the chip transistors had a better f_T . The exact value could not be found because the equipment was limited to 50 MHz, however the computer programs estimated around 50 MHz.

The circuit gain calculated on the computer was fairly close to the lab value. However in this case, the computer actually calculated a better gain, as shown in Table 11.

VIDEO AMPLIFIER (TRANSISTOR)

The DC values, shown in Table 10, were slightly off. This is assumed to be caused by two mismatched transistors, as in the differential amplifier circuit. To verify this, the two transistors were swapped and the voltages were found to be higher rather than lower.

The Transient values, found in Table 13, show that the lab value was slightly larger than the computer calculation. MICRO-CAP III was closer in this case.

The 3 dB frequency point was calculated within 1 MHz by each computer program, as seen in Table 12. The 0 dB point was found to be at 50 MHz in the lab, exactly what MICRO-CAP III calculated and a little under what PSpice calculated.

The gain of the circuit was measured 4 dB lower than what the computer programs estimated, as shown in Table 11. This seems to be an area that is lacking in the computer calculations.

CONCLUSION

Two simple circuits were designed, built, and tested in the lab. These same circuits were also simulated with PSpice and MICRO-CAP III, and the results compared. DC results were very accurate. Transient results were satisfactory for the differential amplifier circuit when the transistor was not dominant ($C_{LOAD} = 300 \text{ pF}$), but when the transistor was dominant ($C_{LOAD} = 5 \text{ pF}$), Transient results were fairly inaccurate. The Transient results were inaccurate for the video amplifier also. The AC results, namely f_{3dB} and f_T , were fairly good if the fact that the chip contained improved transistors was taken into account. Circuit gain seemed to be the greatest deviation point between the lab results and the computer results: as much as a 4 dB difference was found. This is totally unacceptable and a point that must be explored further.

REFERENCES

1. *PSpice*. Microsim Corp., Irvine, Calif., July 1989.
2. *MICRO-CAP III Reference Manual*. Spectrum Software, Sunnyvale, Calif., 1988.
3. Sedra, A. S., and K. C. Smith. *MicroElectronic Circuits*, CBS College Publishing, N.Y., 1987.
4. Beyer, W. H. *CRC Standard Mathematical Tables*, CRC Press, Inc., Fla., 1988.

BIBLIOGRAPHY

Antognetti, P., and G. Massobrio. *Semiconductor Device Modeling with SPICE*. McGraw-Hill Book Company, N.Y., 1988.

Gray, P. R., and R. G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, N.Y., 1984.

MICRO-CAP III Instruction Manual. Spectrum Software, Sunnyvale, Calif., 1988.

Tuinenga, P. W. *A Guide to Circuit Simulation & Analysis Using PSpice*. Prentice-Hall, N.Y., 1988.

NWC TP 7075

Appendix A
TEST SETUP

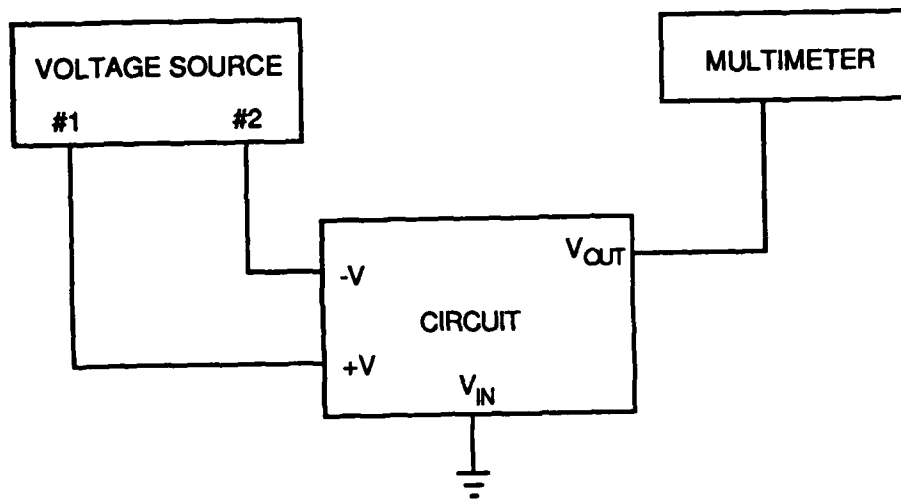


FIGURE A-1. Test Setup to Obtain DC Results.

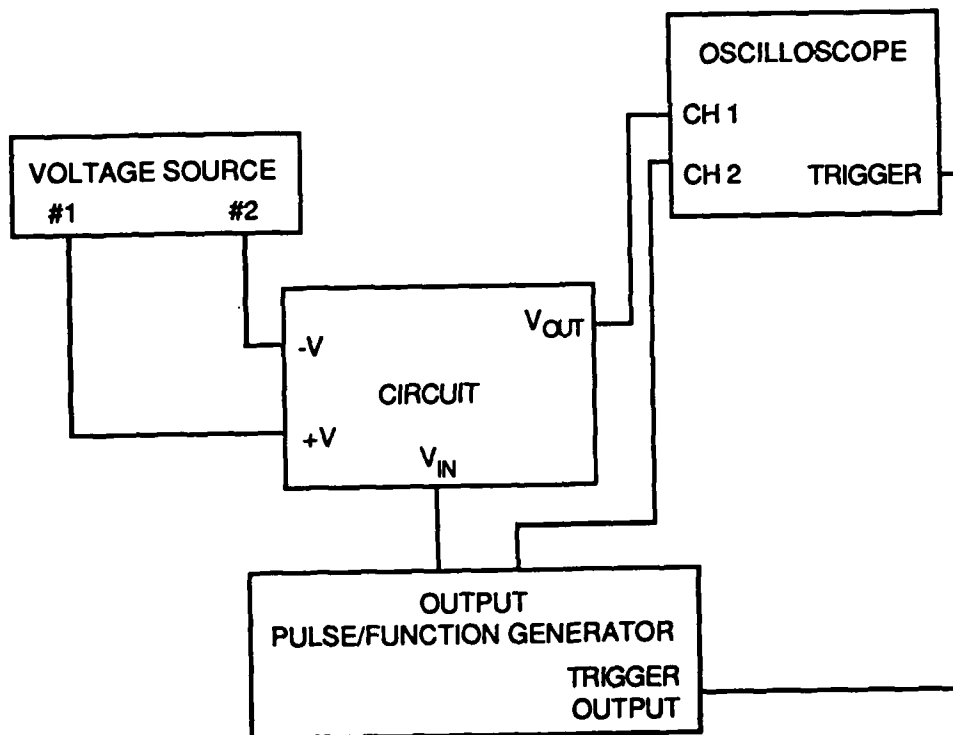


FIGURE A-2. Test Setup to Obtain Transient and AC Results.

NWC TP 7075

Appendix B
PSPICE INPUT FILE

NWC TP 7075

TABLE B-1. Differential Amplifier With Individual Transistors Input File.

DIFFERENTIAL AMPLIFIER BUILT WITH INDIVIDUAL TRANSISTORS

.OPT NOPAGE NOMOD

.WIDTH OUT=80

.TEMP 27

.OP

.DC VIN -0.125 0.125 0.005

.TF V(5) VIN

.AC DEC 10 1K 10MEG

.TRAN 10NS 400NS

.PARAM FACTOR=1.2

VIN 100 0

+ AC 1 0

+ PULSE(0V 50MV 0 0 0 10US 1MS)

VCC 101 0 DC {10*FACTOR}

VEE 102 0 DC {-10*FACTOR}

Q1 4 2 6 Q2N2222A

Q2 5 3 6 Q2N2222A

RS1 2 100 1K

RS2 3 0 1K

RC1 4 101 1K

RC2 5 101 1K

Q3 6 7 102 Q2N2222A

Q4 7 7 102 Q2N2222A

RBIAS 7 101 20K

CLOAD 4 5 5PF

.LIB C:\PSPICE\BIPOLAR.LIB

.PLOT DC V(5)

.PLOT TRAN V(5)

.PLOT AC VDB(5)

.END

NWC TP 7075

TABLE B-2. Differential Amplifier With Transistors
on a Chip Input File.

DIFFERENTIAL AMPLIFIER BUILT WITH CHIP TRANSISTORS

.OPT NOPAGE NOMOD

.WIDTH OUT=80

.TEMP 27

.OP

.DC LIN VIN -0.125 0.125 0.005

.TF V(5) VIN

.AC DEC 20 100KHZ 100MEGHZ

.TRAN 10NS 400NS

.PARAM FACTOR=0.8

VIN 100 0

+ AC 1 0

+ PULSE(0V 50MV 0NS 0NS 0NS 10US 1MS)

VCC 101 0 DC {10*FACTOR}

VEE 102 0 DC {-10*FACTOR}

Q1 4 2 6 Q2N2222A

Q2 5 3 6 Q2N2222A

RS1 2 100 1K

RS2 3 0 1K

RC1 4 101 1K

RC2 5 101 1K

Q3 6 7 102 Q2N2222A

Q4 7 7 102 Q2N2222A

RBIAS 7 101 13K

CLOAD 4 5 5PF

.LIB C:\PSPICE\BIPOLAR.LIB

.PLOT DC V(5)

.PLOT TRAN V(5)

.PLOT AC VDB(5)

.END

NWC TP 7075

TABLE B-3. Video Amplifier Input File.

```

VIDEO AMPLIFIER
.OPT NOPAGE NOMOD
.WIDTH OUT=80
.TEMP 27
.OP
.TF V(9) VIN
.AC DEC 10 10KHZ 100MEGHZ
.TRAN 1NS 100NS
.PARAM FACTOR=.8

VIN 1 0
+ AC 1 0
+ PULSE (0V 30MV 0 0 0 2US .33MS)

VCC 2 0 DC {10*FACTOR}
VEE 12 0 DC {-10*FACTOR}

Q1 3 3 10 Q2N2222A
Q2 5 3 11 Q2N2222A
Q3 2 4 5 Q2N2222A
Q4 6 7 5 Q2N2222A
Q5 2 6 9 Q2N2222A
RBB 2 3 7.15K
RE1 10 12 1K
RE2 11 12 1K
RE3 4 0 1K
RE4 7 0 1K
RC 2 6 3.01K
RF 9 8 4.5K
RX 8 0 .5K
C1 1 4 .1UF
C2 7 8 .1UF

.LIB C:\PSPICE\BIPOLAR.LIB

.PLOT AC VDB(9)
.PLOT TRAN V(9)
.END

```

NWC TP 7075

Appendix C

MICRO-CAP III CIRCUIT DIAGRAM

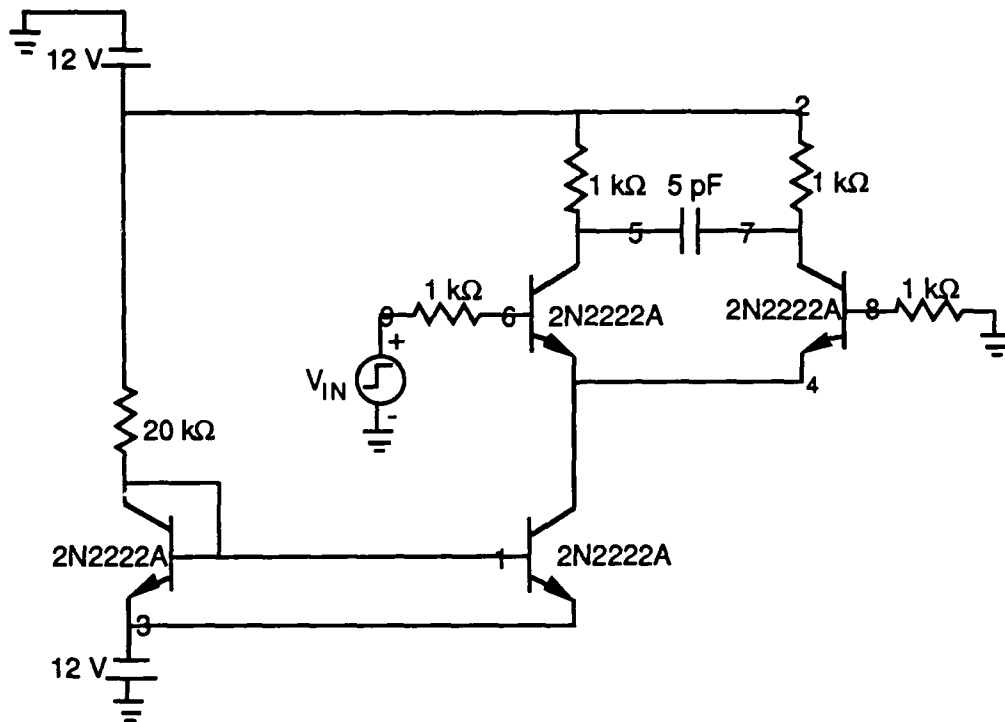


FIGURE C-1. Differential Amplifier With Individual Transistors.

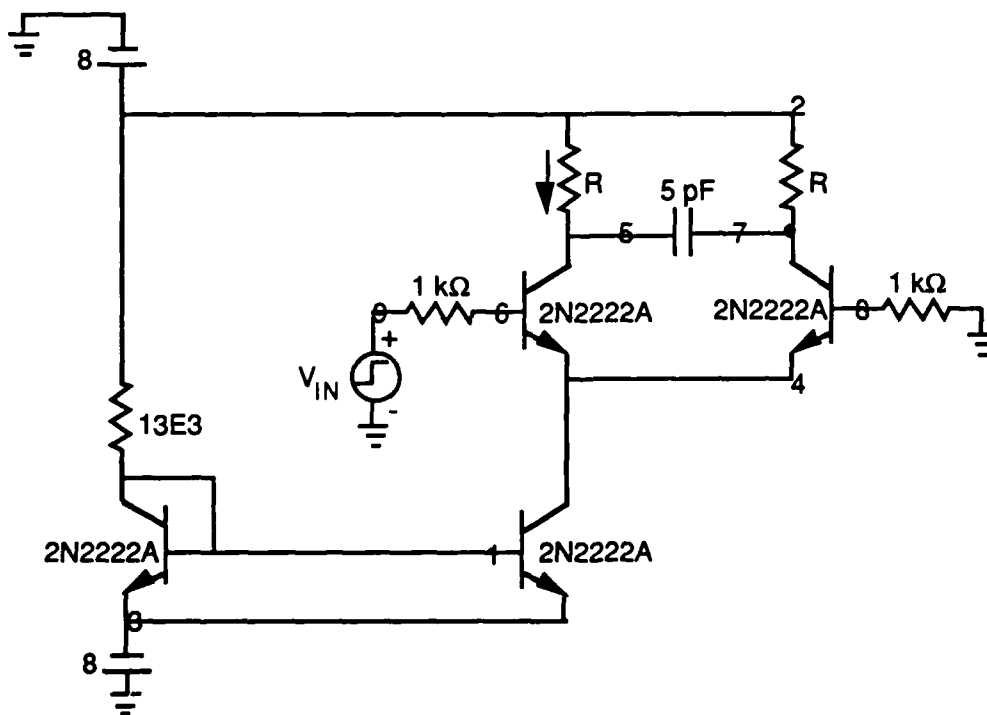


FIGURE C-2. Differential Amplifier With Chip Transistors.

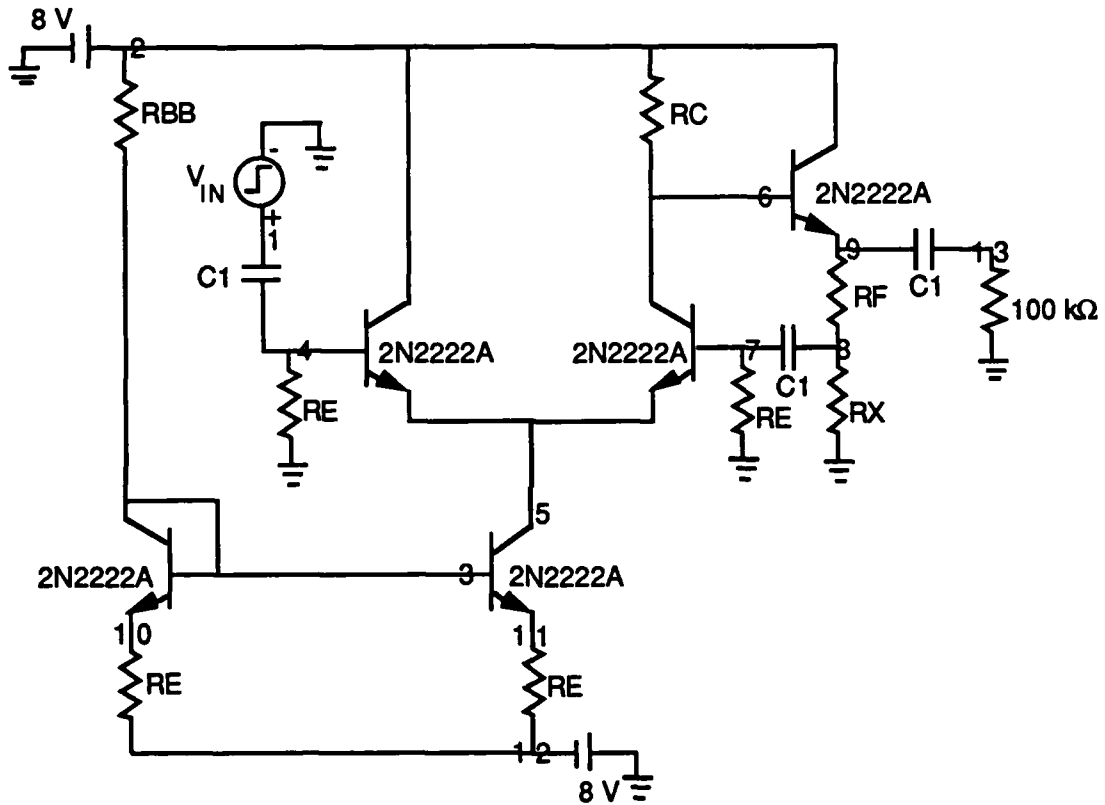


FIGURE C-3. Video Amplifier.

INITIAL DISTRIBUTION

2 Naval Air Systems Command (AIR-5004)
 2 Naval Sea Systems Command (Technical Library)
 1 Commander in Chief, U. S. Pacific Fleet, Pearl Harbor (Code 325)
 1 Commander, Third Fleet, San Francisco
 1 Commander, Seventh Fleet, San Francisco
 2 Naval Academy, Annapolis (Director of Research)
 1 Naval War College, Newport
 1 Air Force Intelligence Agency, Bolling Air Force Base (AFIA/INTAW, Maj. R. Esaw)
 2 Defense Technical Information Center, Alexandria
 1 Hudson Institute, Incorporated, Center for Naval Analyses, Alexandria, VA (Technical Library)

ON CENTER DISTRIBUTION

| | | |
|------------------------------|--------------------|--------------------------------|
| 1 Code 00 | 1 Code 35 | 1 Code 39 |
| 1 Code 01 | 1 Code 3501 | 1 Code 3901A |
| 1 Code 03 | 1 Code 3502 | 1 Code 3902 |
| 1 Code 032 | 1 Code 351 | 1 Code 391 |
| 1 Code 05 | 1 Code 352 | 1 Code 392 |
| 1 Code 06 | 1 Code 35203 | 1 Code 3921 |
| 1 Code 06A | 1 Code 35204 | 1 Code 393 |
| 1 Code 21501 | 1 Code 32405 | 1 Code 394 |
| 1 Code 2504L (Library) | 1 Code 35206 | 1 Code 3943 |
| 1 Code 2721 | 1 Code 3521 | 1 Code 395 |
| 1 Code 3061 | 1 Code 3524 | 1 Code 3952, M. Witham |
| 1 Code 3062 | 1 Code 357 | 1 Code 39B3 |
| 1 Code 31 | 1 Code 3573 | 1 Code 61 |
| 1 Code 3102 | 1 Code 3574 | 1 Code 6102 |
| 1 Code 311 | 1 Code 359 | 1 Code 611 |
| 1 Code 313 | 1 Code 35E | 1 Code 612 |
| 1 Code 314 | 1 Code 36 | 1 Code 613 |
| 1 Code 315 | 1 Code 3602 | 1 Code 614 |
| 1 Code 318 | 1 Code 362 | 1 Code 62 |
| 1 Code 319 | 1 Code 364 | 1 Code 6202 |
| 1 Code 3195 | 1 Code 365 | 1 Code 621 |
| 1 Code 32 | 1 Code 366 | 1 Code 622 |
| 1 Code 3202 | 1 Code 368 | 1 Code 623 |
| 1 Code 321 | 1 Code 3686, GIDEP | 1 Code 625 |
| 1 Code 326 | 1 Code 37 | 1 Code 62C |
| 1 Code 327 | 1 Code 3702 | 1 Code 64 |
| 1 Code 329 | 1 Code 3705 | 1 Code 6402 |
| 1 Code 33 | 1 Code 371 | 1 Code 6403 |
| 1 Code 3303 | 1 Code 372 | 1 Code 641 |
| 1 Code 34 | 1 Code 373 | 1 Code 642 |
| 1 Code 3401 | 1 Code 374 | 1 Code 643 |
| 1 Code 3402 | 1 Code 38 | 1 Code 644 |
| 1 Code 341 | 1 Code 3802 | 1 O.O.D., Administration Bldg. |
| 4 Code 343 (3 plus Archives) | 1 Code 389 | 1 PSD |
| | | 1 VX-5 |